Advances in Engineering Software 45 (2012) 144–152

Contents lists available at SciVerse ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Manufacturer–retailer supply chain coordination: A bi-level programming approach

Ali Naimi Sadigh, Marzieh Mozafari, Behrooz Karimi *

Department of Industrial Engineering and Management Systems, Amirkabir University of Technology, 424 Hafez Ave., Tehran 15916-34311, Iran

ARTICLE INFO

Article history:
Received 19 May 2011
Received in revised form 25 August 2011
Accepted 12 September 2011
Available online 24 October 2011

Keywords:
Manufacturer–retailer supply chain
Stackelberg game
Pricing
Advertising
Modified imperialist competitive algorithm
Evolution strategy

ABSTRACT

This paper investigates a multi-product manufacturer–retailer supply chain where demand of each product is jointly influenced by price and advertising expenditure. We propose a Stackelberg game framework under two power scenarios. In the first, we consider the traditional approach where the manufacturer is the leader. In the latter, we allow the retailer to act as the dominant member of the supply chain. Bi-level programming approach is applied to find the optimal equilibrium prices, advertising expenditures and production policies; then several solution procedures, including imperialist competitive algorithm, modified imperialist competitive algorithm, and evolution strategy are proposed. Finally numerical experiments are carried out to evaluate the effectiveness of models as well as solution procedures.

1. Introduction

A supply chain consists of independent parties which form a chain of process to convert the raw materials into finished products and make them available to ultimate customers. Today intra/inter supply chain competition has given a special attention by researchers, e.g. [1–3]. This paper mainly focuses on vertical inter-chain competition. Increasing competition and market globalization motivate independent firms in different levels of supply chain to coordinate their decisions with the goal of gaining mutual benefit. The two echelon supply chain investigated in this paper represents a single manufacturer which wholesales multiple products to a retailer, who then sells them to the end customers [4–6]. Many articles have studied channel coordination between manufacturer and retailer from different aspects of business decisions, including pricing, advertising, production, and inventory management [7,8]. However studies simultaneously handle more than one aspect of coordination are sparse. For a comprehensive review on channel coordination refer to [9].

Some authors have studied the manufacturer–retailer coordination problem through game theory [10,11]. The condition where each member attempts to maximize his own profit is described as non-cooperative game. In a Stackelberg non-cooperative game, the member with the dominant power (the leader) controls the other members who follow the leader's actions. The leader makes the first move after estimating the reaction of the other members [12]. In manufacturer–retailer supply chains, traditionally manufacturer act as the leader. However, the leading power has shifted from manufacturer to retailer in recent years [13,14].

Reviewing the literature, one can find pricing as a prominent mechanism for supply chain coordination. Eliashberg and Steinberg [15] investigate a two-member channel coordination problem considering production activities such as product delivery and inventory policy, along with pricing. Weng [16] presents a single supplier-multiple buyers coordination model through quantity discounts and franchise fees. He indicates that Stackelberg game can guarantee perfect coordination in such system. Raju and Zhang [17] develop a dominant retailer supply chain model using either quantity discount or two-part tariffs. Yu et al. [18] consider pricing and order intervals decisions in one manufacturer–multiple retailers supply chain. They model the problem as a Stackelberg game where the manufacturer is the leader and retailers are followers.

Several studies on supply chain coordination have considered pricing and advertising decisions simultaneously. Yue et al. [14] address the problem of cooperative advertisement in a manufacturer–retailer supply chain where demand is price sensitive and the manufacturer, as the dominant member, offers price deductions to the retailer. He et al. [19] model a stochastic cooperative advertisement problem through a differential Stackelberg game. Szmerekovsky and Zhang [20] consider pricing and advertising coordination through Stackelberg game in a two-tier supply chain in which demand depends on retail price and advertising expenditure. Two other examples are [8,21]. Esmaeili et al. [22] propose non-cooperative and cooperative games for the seller–buyer coordination to optimize pricing, advertising and inventory decisions while demand non-linearly depends on selling price and marketing

* Corresponding author. Tel.: +98 2164545374.
E-mail addresses: alins@aut.ac.ir (A. Naimi Sadigh), m.mozafari@aut.ac.ir (M. Mozafari), b.karimi@aut.ac.ir (B. Karimi).

0965-9978/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2011.09.008

expenditure. Yu et al. [23] apply similar approach in a VMI (vendor managed inventory) supply chain with one manufacturer and multiple retailers, using a Manufacturer-Stackelberg game model for determining the optimal advertising, pricing and inventory policies.

In this research, we study a multiple-product manufacturer–retailer supply chain where demand is non-linearly influenced by prices and advertising expenditures. Although nonlinear demand functions are more real, they have been rarely used in previous works because of their complexities. We develop two non-cooperative games. The first scenario considers a situation where manufacturer, as the leader of the channel, controls the common production interval and wholesale prices, which are followed by the retailer through optimizing its own selling prices and advertising expenditures. In second scenario, we adopt a similar approach for the situation that the retailer holds the leading power. In addition, we consider budget constraints on production quantity and adverting investment.

The above games are formulated through bi-level optimization. Due to the NP-hard nature of the bi-level models, a solution procedure based on imperialistic competitive algorithm (ICA) is proposed to search the equilibrium solutions. We modify ICA performance through some additional mechanisms. An exhaustive grid search and an evolution strategy (ES) algorithm are also adopted for validation purposes.

The rest of the paper is organized as follows. Section 2 describes the assumptions, objectives, decision variables, constraints and parameters of the manufacturer–retailer supply chain problem. In Section 3, bi-level formulations of the Stackelberg games are developed under the two scenarios. Section 4 proposes solution procedures for solving the Stackelberg games. Section 5 is devoted to experimental results and finally Section 6 discusses the concluding remarks.

2. Problem description

Consider a two member supply chain where the manufacturer wholesales multiple products to the retailer, who then sells them to end customers. We assume that the manufacturer decides on the common production interval T and the unit wholesale prices ψ_i of each product i, $i = 1,2,\ldots,n$. On the other side, the retailer controls the unit retail prices p_i and the advertising expenditures a_i for each product i. The demand $D_i(p_i,a_i)$ for each product i is a joint non-linear function of the retail prices and advertising expenditures as follows (refer to [22]):

$$D_i(p_i,a_i) = k_i \cdot p_i^{-\alpha_i} \cdot a_i^{\beta_i}, \tag{1}$$

where k_i is a positive scaling parameter. $\alpha_i(\alpha_i > 1)$ and $\beta_i(0 < \beta_i < 1,$ $\beta_i + 1 < \alpha_i)$ are the price elasticity and advertising expenditure elasticity, respectively.

We assume a common production interval T in which all the products must be manufactured. This means that all the products have the same production cycle time (i.e. $T = T_i$). Therefore, we can calculate the lot size of product i in each interval as $Q_i = D_i \cdot T$. The manufacturer makes a profit equal to the wholesales revenue minus the production cost, setup and holding costs. Considering parameters A_{s_i} as the unit setup cost for each product i, C_{s_i} as the unit production cost, and ρ as the holding cost percentage per unit, the manufacturer's objective function can be defined as:

$$\begin{aligned}\Pi_s(\psi_i,T) = &\sum_{i=1}^{n}\psi_i \cdot D_i(p_i,a_i) - \sum_{i=1}^{n}C_{s_i} \cdot D_i(p_i,a_i) - \frac{1}{T} \cdot \sum_{i=1}^{n}A_{s_i} \\ &- \frac{1}{2} \cdot \rho \cdot T \cdot \sum_{i=1}^{n}C_{s_i} \cdot D_i(p_i,a_i) \cdot u_i^{-1},\end{aligned} \tag{2}$$

where u_i is a positive constant such that $u_i = r_i/D_i$ and r_i reflects the production rate for product i. We assume that the manufacturer produces Q_i units of product i and then dispatches the whole lot to the retailer. Inasmuch as the manufacturer permits no shortages, the production rate must be at least equal to the expected demand rate (i.e. $\forall i,\ r_i \geqslant d_i$). In such multiproduct manufacturing system, if $\sum_{i=1}^{n}Q_i/r_i \leqslant 1$, then there will exist a feasible time interval, T, in which all products can be manufactured. Thus, we have $\sum_{i=1}^{n}u_i^{-1} \leqslant 1$.

Furthermore, the manufacturer can spend at most B_s units of budget for his production. Thus, he has a constraint defined as follows:

$$\sum_{i=1}^{n}C_{s_i} \cdot T \cdot D_i(p_i,a_i) \leqslant B_s. \tag{3}$$

The retailer determines the selling prices and advertising expenditures in order to maximize his benefit. The retailer's profit can be calculated as the sales revenue minus the purchasing cost, advertising cost, ordering and holding costs, given as follows.

$$\begin{aligned}\Pi_b(p_i,a_i) = &\sum_{i=1}^{n}p_i \cdot D_i(p_i,a_i) - \sum_{i=1}^{n}\psi_i \cdot D_i(p_i,a_i) - \sum_{i=1}^{n}a_i \cdot D_i(p_i,a_i) \\ &- \frac{1}{T} \cdot \sum_{i=1}^{n}A_{b_i} - \frac{1}{2} \cdot \rho \cdot T \cdot \sum_{i=1}^{n}\psi_i \cdot D_i(p_i,a_i),\end{aligned} \tag{4}$$

where parameter A_{b_i} specifies the retailer's unit ordering cost for product i. The holding cost is expressed as a percentage of purchase cost, multiplying by 1/2 to obtain the average inventory value for each product. In addition, the amount of money devoted to advertising must be within a given budget constraint B_b. This constraint can be expressed as follows:

$$\sum_{i=1}^{n}k_i \cdot p_i^{-\alpha_i} \cdot a_i^{\beta_i+1} \leqslant B_b. \tag{5}$$

3. Bi-level programming formulation

Bi-level programming approach provides a framework to deal with situations where a leader firm incorporates within its decision process the reaction of a follower firm to its course of action [24]. Bi-level problems are closely associated with Stackelberg games and Mathematical Programs with Equilibrium Constraints (MPEC), which are both characterized by two levels of optimization problems where the constraint region of the upper level problem is implicitly determined by the lower level optimization problem (refer to [25]). In this section, we model the interactions between the manufacturer and the retailer through bi-level programming under two power scenarios: Manufacturer-Stackelberg and Retailer-Stackelberg.

3.1. The Manufacturer-Stackelberg (MS) model

Here we consider the manufacturer as the leader and the retailer as the follower. The MS model can be formulated as a bi-level program where the manufacturer determines $(\psi_1,\psi_2,\ldots,\psi_n)$ and T at the upper level, subject to his production budget constraint. Then at the lower level, the retailer reacts by choosing optimal prices and advertising policies, based on his own optimization model. Thus, the MS model can be expressed as follows:

Upper-level problem:

$$\begin{aligned}\max_{\psi,T}\Pi_s(\psi_i,T) = &\sum_{i=1}^{n}\psi_i \cdot D_i(p_i,a_i) - \sum_{i=1}^{n}C_{s_i} \cdot D_i(p_i,a_i) - T^{-1} \cdot \sum_{i=1}^{n}A_{s_i} \\ &- \frac{1}{2} \cdot \rho \cdot T \cdot \sum_{i=1}^{n}C_{s_i} \cdot D_i(p_i,a_i) \cdot u_i^{-1}\end{aligned} \tag{6}$$

$$\text{Subject to } \sum_{i=1}^{n}C_{s_i} \cdot T \cdot D_i(p_i,a_i) \leqslant B_s. \tag{7}$$

Lower-level problem:

$$\max_{p,a} \Pi_b(p_i,a_i) = \sum_{i=1}^{n} p_i \cdot D_i(p_i,a_i) - \sum_{i=1}^{n} \psi_i \cdot D_i(p_i,a_i) - \sum_{i=1}^{n} a_i \cdot D_i(p_i,a_i) - T^{-1} \cdot \sum_{i=1}^{n} A_{b_i} - \frac{1}{2} \cdot \rho \cdot T \cdot \sum_{i=1}^{n} \psi_i \cdot D_i(p_i,a_i) \quad (8)$$

$$\text{Subject to } \sum_{i=1}^{n} k_i \cdot p_i^{-\alpha_i} \cdot a_i^{\beta_i+1} \leqslant B_b. \quad (9)$$

The optimal solution of this structure is called the Stackelberg Equilibrium.

3.2. The Retailer-stackelberg (RS) model

The RS model assumes that the retailer holds the manipulative power and acts as the leader. Similar to the previous section, the RS model can be formulated as a bi-level program in which the retailer announces the price vector $(p_1, p_2, \ldots, p_n)$ and the advertising expenditure vector $(a_1, a_2, \ldots, a_n)$ at the upper level and the manufacturer determines the optimal wholesale prices $(\psi_1^*, \psi_2^*, \ldots, \psi_n^*)$ and common production interval T^* in response. Thus, the RS model can be expressed as follows:

$$\max_{p,a} \Pi_b(p_i,a_i) = \sum_{i=1}^{n} p_i \cdot D_i(p_i,a_i) - \sum_{i=1}^{n} \psi_i \cdot D_i(p_i,a_i) - \sum_{i=1}^{n} a_i \cdot D_i(p_i,a_i) - T^{-1} \cdot \sum_{i=1}^{n} A_{b_i} - \frac{1}{2} \cdot \rho \cdot T \cdot \sum_{i=1}^{n} \psi_i \cdot D_i(p_i,a_i) \quad (10)$$

$$\text{Subject to } (\Psi, T) \in \arg\max \Pi_s(\psi_i, T) = \sum_{i=1}^{n} \psi_i \cdot D_i(p_i,a_i) - \sum_{i=1}^{n} C_{s_i} \cdot D_i(p_i,a_i) - T^{-1} \cdot \sum_{i=1}^{n} A_{s_i} - \frac{1}{2} \cdot \rho \cdot T \cdot \sum_{i=1}^{n} C_{s_i} \cdot D_i(p_i,a_i) \cdot u_i^{-1}, \quad (11)$$

$$\sum_{i=1}^{n} k_i \cdot p_i^{-\alpha_i} \cdot a_i^{\beta_i+1} \leqslant B_b, \quad (12)$$

$$\sum_{i=1}^{n} C_{s_i} \cdot T \cdot D_i(p_i,a_i) \leqslant B_s. \quad (13)$$

Due to the nonlinear functions involved in the upper and lower levels problems, both the MS and RS models are NP-hard. To tackle this problem, several solution procedures are proposed in the following section to search the optimal Stackelberg equilibrium solutions of the games.

4. Solution procedures

Consider a bi-level model:

$$\max_{x,y} f(x,y) \quad (14)$$

$$s.t. (x,y) \in X \quad (15)$$

$$y \in S(x), \quad (16)$$

where

$$S(x) = argmax_y g(x,y) \quad (17)$$

$$s.t. (x,y) \in Y. \quad (18)$$

Our solution procedure starts with an initial guess of the optimal upper-level decision value x and moves this initial solution through an exploratory process to achieve a new solution. In each iteration, by solving the lower-level problem, the optimal reaction y^* is obtained and returned to the upper-level model. This procedure continues until an optimal or near-optimal solution is reached for the upper level problem. Fig. 1 depicts the steps of the solution procedure for solving bi-level programs.

In this section, first we propose an ICA algorithm to solve Stackelberg games. Then we develop a Modified ICA (MICA) using some additional mechanisms in order to reach the high quality solutions. An ES algorithm is also applied for comparison purpose.

4.1. ICA method

ICA is a new evolutionary global search algorithm using socio-political process of imperialism and imperialistic competition as a source of inspiration. It has been successfully implemented to a range of optimization problems and shown good performance in both convergence rate and global optima achievement [26–31]. ICA starts with an initial population of individuals, each called a country. Some of the best countries are selected as imperialists and the rest form colonies which are then divided among imperialists based on imperialists' power. After forming the initial empires, competition begins and colonies move towards their relevant imperialists. During competition, weak empires collapse and powerful ones take possession of more colonies. At the end, there exists only one empire while the position of imperialist and its colonies are the same. For a detailed description refer to [26]. In the following, we explain the steps of ICA dealing with MS and RS bi-level models.

4.1.1. Solution representation

Each individual X is called a country and represented as an array of upper level variables. In the MS game, each country is considered as an $n + 1$ dimensions array. The n first positions of the array include wholesale price variables ψ_i, and the last position refers to production cycle T. We initialize the population by generating N_{pop}. The N_{imp} best countries are then chosen as imperialists. The values of ψ_i and T are randomly selected from the intervals $[C_{s_i}, 15C_{s_i}]$ and $[0, B_s/\sum_{i=1}^{n} C_{s_i}]$, respectively. On the other side, in the RS game, each $2 \times n$ dimensions array; the n first positions refer to retail prices p_i and the rest consist advertising expenditures a_i. The initial population involves N_{pop} countries which are generated randomly such that $p_i \in [C_{s_i}, 50C_{s_i}]$ and $a_i \in [0, p_i]$. Fig. 2 illustrates the solution representation for MS and RS games. We assume $N_{pop} = 50 \cdot n$ and $N_{imp} = 10 \cdot n$ for the MS game as well as $N_{pop} = 80 \cdot n$ and $N_{imp} = 8 \cdot n$ for the RS game.

4.1.2. Objective function evaluation and constraint handling

To assess how much a solution is fit to the optimization purpose, a fitness function is evaluated for any candidate solution. In the MS game, for a given solution (individual) of the manufacturer model, the optimal values of the retailer's variables are obtained through solving the lower-level model. Having determined values of the entire variables, we can calculate the fitness value for a given individual as follows:

Step 1. (upper-level problem)
Produce some initial x as solutions of the leader's problem.

Step 2. (lower-level problem)
Optimize the follower's actions for each x and return y^*s to the leader's model.

Step 3. (upper-level problem)
Evaluate the leader's benefit value for each (x,y^*).

Step 4. (upper-level problem)
Move each x to a new position and go to *step 2* until x^* is achieved (a proper stop criterion is met).

Fig. 1. Steps of the solution procedure.

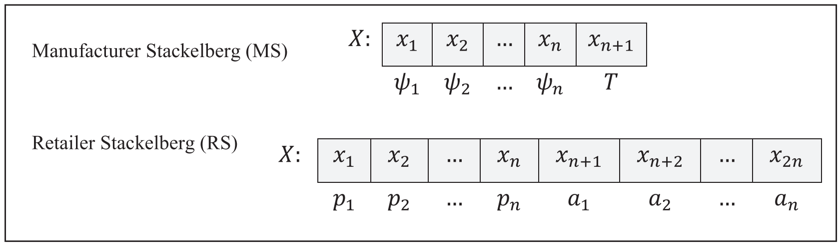

Fig. 2. Solution representation for the MS and RS games.

$$f = \sum_{i=1}^{n}\{\psi_i \cdot D_i(p_i,a_i) - C_{s_i} \cdot D_i(p_i,a_i) - T^{-1} \cdot A_{s_i} - \frac{1}{2} \cdot \rho \cdot T \cdot C_{s_i} \cdot D_i(p_i,a_i) \cdot u_i^{-1}\} - M \cdot \max(0, \sum_{i=1}^{n} C_{s_i} \cdot T \cdot D_i(p_i,a_i) - B_s). \quad (19)$$

The first term in Eq. (19) is the manufacturer's benefit value while the second term penalizes infeasible solutions. The penalty coefficient M is assumed to be a large number. Using penalty term to deal with constraints is a common approach which has been used in many researches, e.g. [32–35].

In the RS game, since the upper-level constraint depends only on self variables, we apply a constraint handling process to ensure the feasibility of a candidate solution. In this way, if $\sum_{i=1}^{n} k_i \cdot p_i^{-\alpha_i} \cdot a_i^{\beta_i+1} - B_b > 0$, we choose a product i randomly and set $p_i = 1.001 \times p_i$ and $a_i = 0.999 \times a_i$. This process continues until the constraint holds. Applying this approach to any candidate solution X, one can find the nearest feasible solution to that candidate solution such that the constraint is often met the whole budget (i.e. $\sum_{i=1}^{n} k_i \cdot p_i^{-\alpha_i} \cdot a_i^{\beta_i+1} - B_b = 0$), which can lead to good solutions.

given a feasible candidate solution of the retailer's model, the manufacturer's model is solved and optimal values of variables are returned to the retailer. The fitness value for each individual in the RS game is calculated as follows:

$$f = \sum_{i=1}^{n}\left\{p_i \cdot D_i(p_i,a_i) - \psi_i \cdot D_i(p_i,a_i) - a_i \cdot D_i(p_i,a_i) - T^{-1} \cdot A_{b_i} - \frac{1}{2} \cdot \rho \cdot T \cdot \psi_i \cdot D_i(p_i,a_i)\right\}. \quad (20)$$

4.1.3. Creation of initial empires

After evaluating the objectives of all individuals, the N_{imp} best countries (countries with greater objective values) are selected as the imperialists. The remained (N_{col}) countries form the colonies which are then divided among the imperialists based on imperialists' power.

The normalized power of the jth imperialist ($j = 1,2,\ldots,N_{imp}$) is defined as

$$P_{imp(j)} = \left|\frac{f_{imp(j)}}{\sum_{i=1}^{N_{imp}} f_{imp(i)}}\right|. \quad (21)$$

Then we can calculate the number of colonies given to the jth imperialist as

$$NOC_{imp(j)} = \mathrm{Round}(P_{imp(j)} \times N_{col}). \quad (22)$$

4.1.4. Assimilation

Assimilation refers to the movement of colonies towards the relevant imperialist. Each colony moves by δ_i units along dimension i, while δ_i is a random variable with uniform distribution $\delta_i \sim U(0, \theta_i \times \Delta_i)$, $\theta_i > 1$ is a escalating parameter and Δ_i is the distance between the colony and the imperialist along dimension i. Fig. 3 illustrates assimilation in a two-dimensional optimization problem.

Here, the parameter θ_i is set as follows:

$$\theta_i = \begin{cases} 2, & i = 1,\ldots,n \\ 1.2, & i = n+1,\ldots,l \end{cases}, \quad (23)$$

where l is equal to $n + 1$ in the MS and $2 \times n$ in the RS problem.

4.1.5. Exchanging the positions of the imperialist and one colony

While moving towards the relevant imperialist, one colony might reach to a position with greater objective value than the imperialist. In this case, the imperialist and the colony change their positions and the algorithm goes on with the new imperialist.

4.1.6. Total power of an empire

The ultimate power of an empire is influenced by the power of its imperialist as well as its colonies. Therefore, the total power of an empire is calculated as follows:

$$TP_{emp(j)} = f_{imp(j)} + \xi \cdot \frac{\sum_{i=1}^{NOC_{imp(j)}} f_{col(i)}}{NOC_{imp(j)}}, \quad (24)$$

where $TP_{emp(j)}$ is the total power of the empire j and ξ is a small positive number. Increasing ξ highlights the effect of colonies. Here we set $\xi = 0.1$, which has shown good results in most of the ICA implementations.

4.1.7. Imperialistic competition

Each empire attempts to take the colonies of other empires under its control. During imperialistic competition, weaker empires are gradually losing their colonies and the powerful ones possess these colonies. Due to this fact, one of the weakest colonies of the weakest empire is selected in every iteration and other empires compete to get the colony. The probability of possession for each empire $POP_{emp(j)}$ is relative to its total power such that

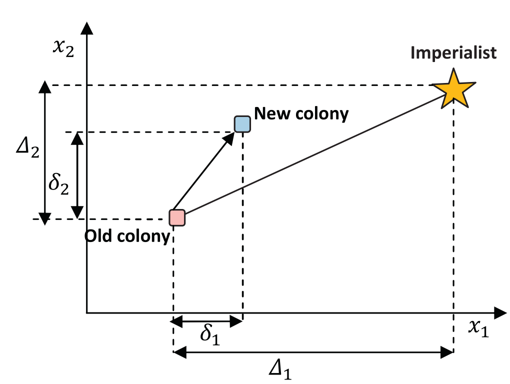

Fig. 3. Movement of colonies toward their relevant imperialist.

$$POP_{emp(j)} = \frac{TP_{emp(j)}}{\sum_{i=1}^{N_{imp}} TP_{emp(i)}}. \tag{25}$$

Let a vector $POP = (POP_1, POP_2, \ldots, POP_{N_{imp}})$, and a vector $R = (R_1, R_2, \ldots, R_{N_{imp}})$, where R_is, $i = 1, 2, \ldots, N_{imp}$, are uniform random numbers from $U(0,1)$. By subtracting R from POP, we can form vector $S = (POP_1 - R_1, POP_2 - R_2, \ldots, POP_{N_{imp}} - R_{N_{imp}})$. Referring to vector S, the selected colony is given to the empire with maximum index in S.

Bringing together all we have discussed, Fig. 4 represents the flowchart of the ICA adopted to solve the Stackelberg games.

4.2. Modified ICA method

The assimilation strategy plays an important role in convergence rate of the ICA as well as the probability of being trapped in local optima. In this section, several mechanisms are proposed in order to improve the performance of the original ICA for solving Stackelberg games.

4.2.1. Normal distributed assimilation

We propose a new assimilation strategy based on normal distribution. According to this strategy, the amount of movement in any iteration is chosen from a normal distribution such that $\delta_i \sim N(0, \Delta_i)$. δ_i denotes the amount of movement for the index i of the colony's array, and Δ_i is the distance between the colony and the imperialist along x_i. The new position of each colony can be obtained from $x_i^{t+1} = x_{imp,i}^t + \delta_i^t$, where $x_{imp,i}^t$ is the value of ith decision variable of the imperialist's array in iteration t and x_i^{t+1} is the updated value of ith decision variable of the colony's array in iteration $t + 1$. Using this approach colonies are moved towards the relevant imperialist searching a symmetric area around the imperialists. In this way, the areas closed to the imperialist are more probable to search than farther areas.

In fact, this assimilation intensifies searching on the imperialist's neighbor. Therefore, some portion of the solution space may be unrewarded during the search process. In order to divert the search to less explored regions, we apply a diversification strategy alongside intensification such that colonies are allowed to move in their self neighbors instead of imperialist's neighbor with a small probability γ. It means that the position of the colony is updated as $x_i^{t+1} = x_i^t + \delta_i^t$ with a probability γ. Here, we found that $\gamma = 0.1$ can result in good convergence to the global optimum.

4.2.2. Adaptive assimilation

To further improve, we use an adaptive controller which adapts the movement vector δ_i based on the progress history. This adaption mechanism suggests that the movement vector should be increased if the success rate (i.e., the proportion of colonies reach to a better position) is high, and it should be decreased if the success rate is low. It can be expressed as follows:

$$d_i^{t+1} = \left\{ \begin{array}{ll} c_{dec} \cdot \delta_i^{t+1} & P_s(t) < \iota \\ \delta_i^{t+1} & \iota \leqslant P_s(t) \leqslant \kappa \\ \delta_i^{t+1}/c_{inc} & P_s(t) > \kappa \end{array} \right\}, \tag{26}$$

where $P_s(t)$ defines the success rate at iteration t, δ_i^{t+1} denotes the movement step size at iteration $t + 1$, c_{dec} and c_{inc} are damping parameters which control the changes in movement vector. Adaptive assimilation can enhance the ability of escaping from local optima and fast converging to global optimum. Table 1 represents best values for the parameters for solving MS and RS games.

4.3. ES method

A well-known ES algorithm is also implemented and evaluated for the solution of Stackelberg models. ES is a nature-inspired optimization method which was originally developed by Rechenberg [36] and Schwefel [37] to optimize real-value engineering design problems. ES applies a stochastic iterative procedure to evolve a population of individuals over a selected number of generations. Readers are referred to [38–42] for more examples of ES. The basic concepts used for solving the models through ES is similar to the procedure followed with ICA. Hence, we only focus on the differences. The initial population consists of N_{pop} individuals; each individual defined by $X = (x_i, \sigma_i)$, where x_is denote the upper level problem variables and σ_is are standard deviations used for the

Table 1
Parameters values set for adaptive assimilation.

Parameter	Value	
	Manufacturer Stackelberg	Retailer Stackelberg
ι	1/5	1/5
κ	3/5	4/5
c_{dec}	0.8 + 0.2*rand ( )	0.7 + 0.3*rand ( )
c_{inc}	0.1 + 0.4*rand ( )	0.1 + 0.6*rand ( )

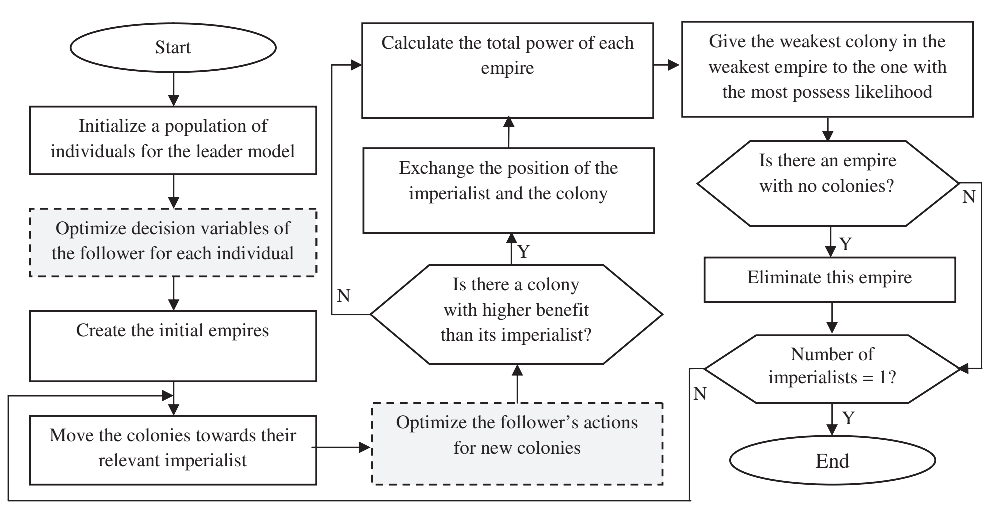

Fig. 4. Flowchart of the ICA algorithm proposed for solving the Stackelberg games.

mutation operation. All individuals are randomly generated and then mutated. A mutated offspring $X' = (x_i', \sigma_i')$ is obtained from a parent X, such that σ_i' and x_i' are defined as follows:

$$\sigma_i' = \sigma_i \cdot \exp(\tau' \cdot z + \tau \cdot z_i), \quad (27)$$

$$x_i' = x_i + \sigma_i' \cdot z_i, \quad (28)$$

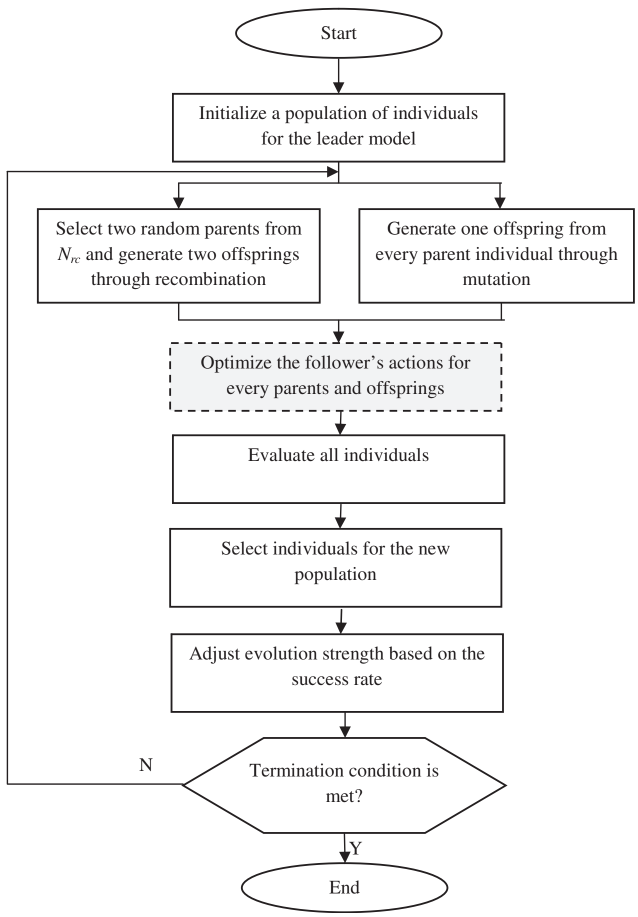

Fig. 5. Flowchart of the ES algorithm applied for the Stackelberg games.

where z and z_i are chosen from the standard normal distribution. τ' denotes a global coefficient called overall learning rate and τ denotes a local coefficient called coordinating wise learning rate. Referring to [39], these coefficients are valued as follows:

$$\tau' = \left(2\sqrt{l}\right)^{-1}, \qquad \tau = \left(\sqrt{2\sqrt{2l}}\right)^{-1}, \quad (29)$$

where l defines the number of decision variables x_i.

The recombination operator selects N_{rc} parent individuals and shares the information such that every two random parents of N_{rc} produce two new offsprings. Consider two parents p_1 and p_2, the offsprings $X' = (x_i', \sigma_i')$ and $X'' = (x_i'', \sigma_i'')$ are obtained through linear recombination of the parents as follows:

$$\sigma_i' = \lambda \cdot \sigma_i^{p_1} + (1-\lambda) \cdot \sigma_i^{p_2} \quad \text{and} \quad \sigma_i'' = (1-\lambda) \cdot \sigma_i^{p_1} + \lambda \cdot \sigma_i^{p_2}, \quad (30)$$

$$x_i' = \lambda \cdot x_i^{p_1} + (1-\lambda) \cdot x_i^{p_2} \quad \text{and} \quad x_i'' = (1-\lambda) \cdot x_i^{p_1} + \lambda \cdot x_i^{p_2}, \quad (31)$$

where λ is a random number from $U(0,1)$. Since both the algorithm's performance and success probability depend on σ, we apply the 1/5-success rule to control this parameter. Let P_s is the ratio of the number of successes to the total number of trials. The 1/5-success rule states that the mutation strength must be reduced if $P_s < 1/5$, whereas in the opposite case $P_s < 1/5$, σ must be increased. It can be expressed by:

$$\sigma_i^{t+1} = \begin{Bmatrix} c_{dec} \cdot \sigma_i^{t+1} & P_s(t) < 1/5 \\ \sigma_i^{t+1} & P_s(t) = 1/5 \\ \sigma_i^{t+1}/c_{inc} & P_s(t) > 1/5 \end{Bmatrix}, \quad (32)$$

where $P_s(t)$ is the success rate, c_{dec} and c_{inc} are the decreasing factor and increasing factor, respectively. σ_i^{t+1} defines the deviation value of ith decision variable in generation $t + 1$ and σ_i^{t+1} is the updated deviation value of ith decision variable in the next generation. To avoid zero value for standard deviation, whenever σ turns into a value less than 0.01, σ is fixed to 0.01. We found that $c_{dec} = 0.7 + 0.3 * rand$ and $c_{inc} = 0.1 + 0.9 * rand$ in the MS game, $c_{dec} = 0.8$ and $c_{inc} = 0.1 + 0.6 * rand$ in the RS game, can lead to best results.

After performing constraint handling process, we evaluate each $X = (x_i, \sigma_i)$ by its objective function and sort them. In order to select individuals for the next generation, we choose best N_{rc} individuals

Table 2
Results on test problems for the RS game.

Problem	n	Retailer's benefit				GAP(%)		Time (s)		
		Grid search	MICA	ICA	ES	MICA-ICA	MICA-ES	MICA	ICA	ES
TEST 1	1	3837	3840.2	3840.2	3840.2	0	0	93	25	253
	2	7607.5	7618.1	7618.1	7618.1	0	0	152	53	327
	3	8649.3	8728.8	8718.4	8726.3	0.119	0.029	197	72	405
	4	–	11748.8	11718.8	11740.0	0.256	0.075	259	96	496
	5	–	15521.9	15487.0	15510.5	0.225	0.073	314	121	563
	6	–	16406.2	16315.2	16390.4	0.558	0.096	374	159	695
	7	–	17835.5	17709.8	17817.6	0.71	0.100	436	174	819
	8	–	19573.2	19396.1	19532.8	0.913	0.207	471	205	863
	9	–	21763.4	21571.6	21725.9	0.889	0.173	513	228	997
	10	–	23782.6	23434.8	23705.5	1.484	0.325	577	260	1163
TEST 2	1	1299.1	1300.7	1300.7	1300.7	0	0	102	26	241
	2	1628.3	1633.0	1633.0	1633.0	0	0	146	49	338
	3	1312.1	3964.2	3957.6	3962.9	0.167	0.033	208	74	417
	4	–	4687.7	4672.8	4684.2	0.319	0.075	261	99	512
	5	–	5902.1	5874.9	5897.2	0.463	0.083	305	126	597
	6	–	7255.1	7182.1	7243.7	1.016	0.157	363	154	705
	7	–	7589.1	7522.6	7580.4	0.884	0.115	412	181	786
	8	–	8339.3	8247.8	8322.4	1.109	0.203	489	217	929
	9	–	9959	9833.8	9936.6	1.273	0.225	537	231	1045
	10	–	10963.7	10816.5	10932.8	1.361	0.283	596	257	1314

and $(N_{pop} - N_{rc})$ individuals are then selected randomly from the remaining. The algorithm stops if the iteration counter reaches the maximum number of iterations t_{max}. We set $t_{max} = 200 \cdot n$, $N_{pop} = 10 \cdot n$ and $N_{rc} = 4 \cdot n$. The flowchart of the ES algorithm for solving the Stackelberg games is shown in Fig. 5.

5. Computational experiments

In this section numerical experiments are developed to evaluate the ability of the solution procedure to find the optimum equilibrium points of MS and RS games. We test the proposed approaches on a range of problems and summarize the results.

Test problems of different sizes are generated randomly using the same concepts as employed in [22]. It is assumed that the manufacturer produces a number of products from the range of $n \in \{1,2,\ldots,10\}$ and two random problems are generated for any members of this set. we arrange the test problems into two classes, TEST 1 and TEST 2. The elasticity coefficients β_i and α_i are respectively selected from uniform distributions $U(0.05, 0.97)$ and $U(1.5, 3)$ such that $\alpha_i > \beta_i + 1$, and the escalating parameterk_i is chosen from $U(15{,}000, 125{,}000)$. The other parameters of MS and RS games are generated as: $A_{s_i} \sim U(140, 700), C_{s_i} \sim U(1.5, 8.5)$, $A_{b_i} \sim U(40, 500)$, and $\rho = 0.1$.

An exhaustive grid search within the domain of interest is carried out to find near optimum solutions for small instances of size $n = 1$ to $n = 3$ (and also $n = 4$ in the MS case) while for larger instances grid search is not possible given the average computational resources currently available. Grid search yields a very good lower bound for the benefit functions, which can be employed to validate the results of our solution algorithms. In addition, we compare the results of the MICA algorithm with those obtained by the original ICA and ES algorithms. Experiments are performed on TEST 1 and TEST 2 instances under two scenarios of the MS and RS games. The results are summarized in Tables 2 and 3. All algorithms have been implemented in MATLAB 7.6.0 and tested on a Core 2 Duo 2.26 GHz processor with 2 GB of main memory. For optimization

Table 3
Results on test problems for the MS game.

Problem No.	n	Manufacturer's benefit				GAP(%)		Time (s)		
		Grid search	MICA	ICA	ES	MICA-ICA	MICA-ES	MICA	ICA	ES
TEST 1	1	914.7	915.3	915.3	915.3	0	0	44	11	149
	2	1066.5	1070.6	1070.6	1070.6	0	0	87	35	231
	3	1210.5	1231.7	1231.7	1231.7	0	0	127	56	316
	4	1901.6	1940.1	1937.3	1938.3	0.146	0.093	170	73	402
	5	–	2485.7	2476.1	2478.7	0.389	0.282	219	91	471
	6	–	2716.9	2705.1	2708.2	0.438	0.321	274	115	548
	7	–	3024.0	3007.0	3014.6	0.564	0.312	317	131	619
	8	–	3219.8	3183.8	3204.6	1.13	0.474	365	169	705
	9	–	3574.5	3537.8	3558.3	1.037	0.455	403	181	784
	10	–	3844.3	3795.3	3821.6	1.291	0.594	438	205	831
TEST 2	1	207.3	207.7	207.7	207.7	0	0	39	9	137
	2	282.4	284.5	284.5	284.5	0	0	82	31	225
	3	613.1	627.5	627.5	627.5	0	0	135	59	318
	4	876.4	900.8	899.1	899.3	0.200	0.167	184	77	412
	5	–	1333.5	1327.6	1328.9	0.445	0.346	211	87	469
	6	–	1479.0	1474.2	1475.8	0.323	0.217	266	109	541
	7	–	1640.5	1625.0	1631.3	0.955	0.564	311	135	610
	8	–	1765.3	1741.0	1752.8	1.395	0.713	353	162	696
	9	–	1979.5	1954.6	1968.0	1.273	0.584	395	176	773
	10	–	2196.9	2165.8	2183.2	1.436	0.627	432	201	826

Table 4
Comparison between the manufacturer and retailer benefit under two Stackelberg scenarios for test problems.

Problem	n	Retailer's benefit		Manufacturer's benefit	
		Retailer Stackelberg	Manufacturer Stackelberg	Retailer Stackelberg	Manufacturer Stackelberg
TEST 1	1	3840.2	2638.3	682.4	915.3
	2	7618.1	5819.9	802.0	1070.6
	3	8728.8	6301.5	904.1	1231.7
	4	11748.8	8273.5	1414.2	1940.1
	5	15521.9	10658.2	1504.1	2485.7
	6	16406.2	10906.3	1685.9	2716.9
	7	17835.5	11805.3	1709.8	3024.0
	8	19573.2	12777.5	1932.5	3219.8
	9	21763.4	14640.7	1964.5	3574.5
	10	23782.6	15753.9	2168.4	3844.3
TEST 2	1	1300.7	836.8	205.4	207.7
	2	1633.0	1194.6	251.1	284.5
	3	3964.2	2735.6	587.2	627.5
	4	4687.7	3283.9	726.2	900.8
	5	5902.1	4598.0	1076.5	1333.5
	6	7255.1	5347.9	1147.3	1479.0
	7	7589.1	5376.3	1214.4	1640.5
	8	8339.3	6169.5	1479.8	1765.3
	9	9959	6675.4	1512.6	1979.5
	10	10963.7	7640.4	1766.2	2196.9

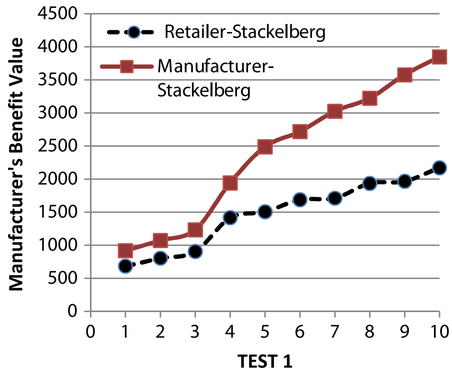

Fig. 6. Manufacturer's benefit comparisons for TEST 1.

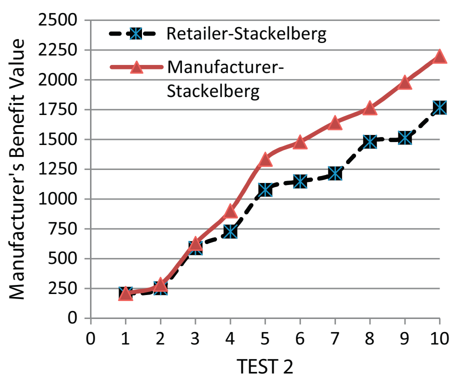

Fig. 7. Manufacturer's benefit comparisons for TEST 2.

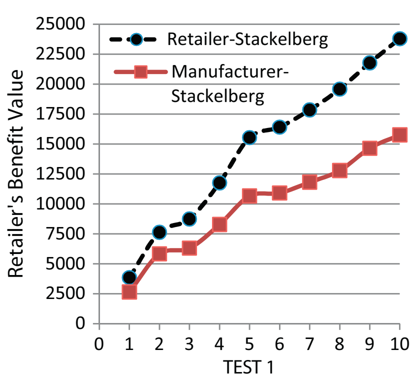

Fig. 8. Retailer's benefit comparisons for TEST 1.

of the lower-lever (follower) problem in each generation of the solution algorithms, we used the GAMS 29.9.2 software interface.

Experimental results indicate that the proposed MICA algorithm exceeds not only the original ICA but also the grid search and ES algorithm in finding the optimal (or near-optimal) equilibrium solutions of the MS and RS games. Concerning computational time, ICA is the most efficient algorithm. However, the proposed MICA algorithm requires a reasonable amount of time and computational efforts such that in the largest instances it takes 600 s which is significantly lower than the amount of time needed by ES algorithm.

Since the benefit value is the most important performance measure in supply chains, we compare the optimal benefit values of the manufacturer and the retailer under two Stackelberg scenarios in Table 4.

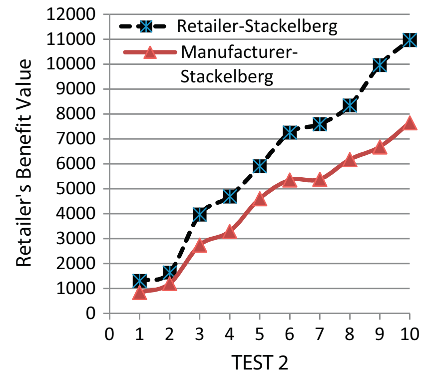

Fig. 9. Retailer's benefit comparisons for TEST 2.

Results are displayed in Figs. 6–9. As it is illustrated, each channel member gains more benefit when playing the Stackelberg leader at the expense of the other channel member who becomes the follower. Finally, it can be concluded that the market power of each member which specifies its decision behavior in the supply chain.

6. Concluding remarks

This paper has studied a multi-product manufacturer–retailer supply chain where the demand jointly depends on price and advertising expenditure. A Stackelberg game framework has developed under two power scenarios. The MS game scenario wherein the manufacturer has the leading power of the chain, and the RS game scenario which allows the retailer to acts as the dominant member in the chain. We have formulated models using bi-level optimization approach to find the optimal equilibrium wholesale and retail prices as well as advertising expenditures and production policies.

Several solution procedures including an ICA and ES methods have been designed to solve Stackelberg games. We proposed a modified version of ICA (MICA) applying some additional mechanisms including a modified assimilation strategy along with a diversification approach, and an adaptive mechanism, to improve the performance of the algorithm. Numerical experiments were carried out for validation and evaluation purposes. Comparing the results with a good lower bound obtained from an exhaustive grid search indicates that all the proposed solution algorithms are able to find high quality solutions in a reasonable amount of time while the MICA exceeds other algorithms in achieving the optimal equilibrium solutions.

References

[1] Edward JA, Yong B. Price competition with integrated and decentralized supply chains. Eur J Oper Res 2010;200:227–34.
[2] Rezapour S, Zanjirani Farahani R. Strategic design of competing centralized supply chain networks for markets with deterministic demands. Adv Eng Software 2010;41:810–22.
[3] Santana S, Sarmah SP. Coordination and price competition in a duopoly common retailer supply chain. Comput Ind Eng 2010;59:280–95.
[4] Yang SL, Zhou YW. Two-echelon supply chain models: considering duopolistic retailers' different competitive behaviors. Int J Prod Econ 2006;103:104–16.
[5] Chen MS, Chang HJ, Huang CW, Liao CN. Channel coordination and transaction cost: a game-theoretic analysis. Ind Market Manag 2006;35:178–90.
[6] Dai Y, Chao X, Fang SC, Nuttle HLW. Pricing in revenue management for multiple firms competing for customers. Eur J Oper Res 2005;98:1–16.
[7] Sarmah SP, Acharya D, Goyal SK. Buyer vendor coordination models in supply chain management. Eur J Oper Res 2006;175(1):1–15.
[8] Xie J, Wie JC. Coordinating advertising and pricing in a manufacturer–retailer channel. Eur J Oper Res 2009;197:785–91.
[9] Fugate B, Sahin F, Mentzer J. Supply chain management coordination mechanisms. J Business Logis 2006;27(2):129–61.
[10] Leng M, Parlar M. Game-theoretic applications in supply chain management: a review. INFOR Inform Syst Oper Res 2005;43(3):187–220.

[11] Cachon G, Netessine S. Game theory in supply chain analysis. In: Simchi-Levi EbD, Wu SD, Shen M, editors. Supply chain analysis in the e-business era. Norwell, MA: Kluwer Academic; 2004.
[12] Gaski JF. The theory of power and conflict in channels of distribution. J Market 1984;15:107–11.
[13] Huang Z, Li SX, Mahajan V. An analysis of manufacturer–retailer supply chain coordination in cooperative advertising. Decision Sci 2002;33:469–94.
[14] Yue J, Austin J, Wang MC, Huang Z. Coordination of cooperative advertising in a two-level supply chain when manufacturer offers discount. Eur J Oper Res 2006;168:65–85.
[15] Eliashberg J, Steinberg R. Marketing–production decisions in an industrial channel of distribution. Manag Sci 1987;33(8):981–1000.
[16] Weng ZK. Channel coordination and quantity discounts. Manag Sci 1995;41(9):1509–22.
[17] Raju J, Zhang ZJ. Channel coordination in the presence of dominate retailer. Market Sci 2005;24(2):254–62.
[18] Yu Y, Liang L, Huang GQ. Leader–follower game in vendor-managed inventory system with limited production capacity considering wholesale and retail prices. Int J Logis: Res Appl 2006;9(4):335–50.
[19] He X, Prasad A, Sethi S. Cooperative advertising and pricing in a dynamic stochastic supply chain: feedback Stackelberg strategies. Prod Oper Manag 2009;18(1):78–94.
[20] Szmerekovsky J, Zhang J. Pricing and two-tier advertising with one manufacturer and one retailer. Eur J Oper Res 2009;192(3):904–17.
[21] Xie J, Neyret A. Co-op advertising and pricing models in manufacturer–retailer supply chains. Comput Ind Eng 2009;56(4):1375–85.
[22] Esmaeili M, Aryanezhad Mir-Bahador, Zeephongsekul P. A game theory approach in seller–buyer supply chain. Eur J Oper Res 2009;191(2):442–8.
[23] Yu Y, Huang GQ, Liang L. Stackelberg game-theoretic model for optimizing advertising, pricing and inventory policies in vendor managed inventory (VMI) production supply chains. Comput Ind Eng 2009;57:368–82.
[24] Marcotte P, Savard G. Bilevel programming: a combinatorial perspective, graph theory and combinatorial optimization. In: Avis D, Hertz A, Marcotte O, editors. Boston: Kluwer Academic Publishers; 2005. p. 191–218.
[25] Colson B, Marcotte P, Savard G. An overview of bilevel optimization. Ann Oper Res 2007;153:235–56.
[26] Atashpaz-Gargari E, Lucas C. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. IEEE Congr Evol Comput 2007:4661–7.
[27] Nazari-Shirkouhi S, Eivazy H, Ghodsi R, Rezaie K, Atashpaz-Gargari E. Solving the integrated product mix-outsourcing problem using the imperialist competitive algorithm. Expert Syst Appl 2010;37(12):7615–26.
[28] Rajabioun R, Atashpaz-Gargari E, Lucas C. Colonial competitive algorithm as a tool for Nash equilibrium point achievement. Comput Sci Appl – ICCSA 2008:680–95.
[29] Duan H, Xu C, Liu S, Shao S. Template matching using chaotic imperialist competitive algorithm. Pattern Recogn Lett 2009;39(6):1362–81.
[30] Jolai F, Sangari M, Babaie M. Pareto simulated annealing and colonial competitive algorithm to solve an offline scheduling problem with rejection. J Eng Manuf 2010;224(7):1119–31.
[31] Atashpaz-Gargari E, Hashemzadeh F, Rajabioun R, Lucas C. Colonial competitive algorithm: a novel approach for PID controller design in MIMO distillation column process. Int J Intell Comput Cybern 2008;1(3):337–55.
[32] Naimi Sadigh A, Mozafari M, Husseinzadeh Kashan A. Mixed integer linear program and Tabu search approach for the complementary edge covering problem. Adv Eng Software 2010;41:762–8.
[33] Wu WH, Lin CY. The second generation of self-organizing adaptive penalty strategy for constrained genetic search. Adv Eng Software 2004;35:815–25.
[34] Wah BW, Chen Y. Constraint partitioning in penalty formulations for solving temporal planning problems. Artif Intell 2006;170:187–231.
[35] Cao D, Chen M. Using penalty function and Tabu search to solve cell formation problems with fixed cell cost. Comput Oper Res 2004;31:21–37.
[36] Rechenberg I. Evolutionstrategie. Stuttgart: Fromman-Holzboog; 1973.
[37] Schwefel HP. Numeriche optimierung von computer-modellen mittels der evolutions-strategie. Basel: Birkauser; 1977.
[38] Whitley D. An overview of evolutionary algorithms: practical issues and common pitfalls. Inform Software Technol 2001;43:817–31.
[39] Beyer HG, Schwefel HP. Evolution strategies: a comprehensive introduction. Nat Comput 2002;1:3–52.
[40] Ahn HJ. Evaluating customer aid functions of online stores with agent-based models of customer behavior and evolution strategy. Inform Sci 2010;180:1555–70.
[41] Chen TY, Hsu YS. A multiobjective optimization solver using rank-niche evolution strategy. Adv Eng Software 2006;37:684–99.
[42] Ebenau C, Rotsschafer J, Thierauf G. An advanced evolutionary strategy with an adaptive penalty function for mixed-discrete structural optimisation. Adv Eng Software 2005;36:29–38.