Int J Adv Manuf Technol
DOI 10.1007/s00170-011-3820-6

ORIGINAL ARTICLE

The economic lot scheduling problem with deteriorating items and shortage: an imperialist competitive algorithm

V. Kayvanfar · M. Zandieh

Received: 19 October 2010 / Accepted: 28 November 2011

Abstract This paper addresses an economic lot scheduling problem (ELSP) for manufacturing environments regarding slack costs and deteriorating items using the extended basic period approach under Power-of-Two (PoT) policy. The purpose of this research is to determine an optimal batch size for a product and minimizing total related costs to such a problem. The cost function consists of three components, namely, setup cost, holding cost includes deteriorating factor, and slack cost. The ELSP is concerned with the scheduling decision of *n* items and lot sizing. Avoiding schedule interference is the main problem in ELSP. The used PoT policy ensures that the replenishment cycle of each item to be integer and this task reduces potential schedule interferences. Since the ELSP is shown as an NP-hard problem, an imperialist competitive algorithm is employed to provide good solutions within reasonable computational times. Computational results show that the proposed approach can efficiently solve such complicated problems.

Keywords Economic lot scheduling problem · Deterioration factor · Shortage cost · Imperialist competitive algorithm

V. Kayvanfar
Department of Industrial Engineering,
Mazandaran University of Science and Technology,
Babol, Iran

M. Zandieh (✉)
Department of Industrial Management, Management and Accounting Faculty, Shahid Beheshti University, G.C.,
Tehran, Iran
e-mail: mostafazzz@yahoo.com

V. Kayvanfar
Research Institute of Food Science and Technology,
P.O. Box 91735-139, Mashhad, Iran

1 Introduction

Scheduling the production of several products on a single facility with the objective of reducing the sum of holding costs and setup costs has been studied in the literature as formal analysis since 1950s as economic lot scheduling problem (ELSP). The ELSP merges lot sizing and production scheduling decisions and is one of the most representative topics. The optimal solution to this problem is known to be quite difficult. The conventional ELSP is concerned with the scheduling of cyclical production of two or more than two products on a single facility in which lots are different in size and consequently are different in production times and cycles, over an infinite planning horizon, assuming deterministic demand for each product. On the other hand, the conventional ELSP is defined as the problem of finding the production sequence, production times, and idle times of several products in a single facility in a cyclic schedule so that the demands are made without stock-outs or backorders and average inventory holding and setup costs are minimized [31].

In this research, we present a model including deterioration factor and slack cost. In real world and in many industries, deterioration occurs for a lot of items such as food industries. Soman et al. [37] showed that in these industries, products do have limited shelf life that restricts the amount of inventory that can be carried without spoilage. Actually, products must have limited storing life because of decreasing the quality of products or deterioration.

The ELSP occurs when one machine is used to meet deterministic and fixed demand of several products over an infinite horizon. Also, the issue of batching arises because the system usually requires a setup cost and a setup time when a

Published online: 21 January 2012

machine switches from one product to the next. In addition to the discrete parts manufacturing, multi-products or multi-purpose processors are common features in many chemical plants such as those producing pharmaceuticals, biochemical, polymers, cosmetics, food, and beverages, etc. Therefore, any methodology for solving the ELSP has huge potential of applicability for industry [21]. Of another potential, lot sizing applications can be applied to high technology industries. Storerooms, which employ automated storage and retrieval systems, supply numerous electronic components, in a specified mix, to circuit-board assembly lines [27]. The raw material is a further usage of ELSP which leads to inventory holding costs. In most cases, the raw materials represent a major part of the cost of the finished product and forsaking these can lead to ineffective policies; for instance, injection molding of plastic parts, such as panels and fixtures in automobiles, refrigerators, or consumer electronics. Another example in such area arises where the raw material is produced internally at a finite rate in the packaging of liquid medical products. The pharmaceutical industry employs common machines to bottle and package different products, and the bottled products are the finished items [17].

To the best of the authors' knowledge, there is no other study that directly survey the conventional ELSP model that includes both the deterioration factor and slack cost; however, Yao and Huang [43] presented an ELSP model including deteriorating factor. A production plan in the ELSP schedules the items within “basic periods,” where a basic period (BP) is an interval of time that consists of setup and production of a subset (or all) of the products [41]. The solution of the ELSP is usually given in terms of a set of multipliers $\{k_i\}$; i=1, 2,…, n and the BP in which each product is produced. The BPs in the ELSP can be categorized as either the “BP” or the “extended basic period” (EBP) approach. The BP approach assumes that the production runs of all products must be made in each BP and this BP must be long enough to accommodate the production of all the products. The researchers have demonstrated that the ELSP under the EBP approach, denoted as the ELSP (EBP), always yields better solutions [14]. In the literature, changing the policy of resolving from the BP approach to the EBP approach is for eliminating the wasted capacity of the production facility due to the restrictive feasibility condition. Using of this approach causes that the BPs are not equally loaded. Consequently, minimizing the maximal load on any BP is another objective as well as the feasibility and the objective of minimizing the total operating cost, because it leads to smaller BPs and consequently lower cost.

Since Hsu [22] has shown that even in the absence of setup costs this problem is NP-hard, if for instance, one employs some branch-and-bound (*B&B*) algorithm or other deterministic and optimum algorithms, for solving such scheduling problem, the search could be computationally expensive since there may be many intermediate sets of basic periods and their corresponding multipliers *K(B)* that need to be tested for feasibility, or for which one must obtain a feasible production schedule. Now there is a problem, and that is misinforming the search by using some greedy algorithms which is caused by quality of the obtained solutions in solving such scheduling problem which consequently search process mislead by misjudging the feasibility of an intermediate set of *K(B)* at a particular BP. So, a “reliable heuristic” that compromises between optimality and computational efficiency is needed.

The power-of-two (PoT) policy necessitates that $k_i=2^q$; $q\geq 0$ integer, for all k_i in the set of multipliers *K(B)*. On the other hand, the multipliers are restricted to be PoT. This policy recently became popular for lot scheduling problems because it reduces potential interferences. There are a lot of reasons in support of the acceptance of the PoT policy. Under PoT policy, researchers were able to derive some easy and effective heuristics to solve both incapacitated and capacitated lot sizing problems. It is interesting that the worst case bounds for PoT policy are actually reasonably tight [41]. Also, implementing the replenishment strategies under PoT policy for decision makers is easier than other policies.

In this paper, an imperialist competitive algorithm (ICA) is proposed to Slack-Deter-ELSP (EBP-PoT). The paper has following structure. Section 2 gives literature review of conventional ELSP with and without different policies. Section 3 includes problem description. Section 4 introduces the proposed ICA. Section 5 explains genetic algorithm (GA) implementation for our problem. Section 6 describes how to generate a feasible solution to our problem. Section 7 presents experimental design. Section 8 includes experimental results achieved by proposed ICA which have been compared those achieved by past GA. Finally, Section 9 consists of conclusions and future work.

2 Literature review

In this section, we will review the literature of ELSP with and without different kinds of policies.

As pointed out in previous section, ELSP is concerned with the scheduling of the cyclical production of $n\geq 2$ items on a single facility in batches that are different in size and consequently different in production time and cycle. In ELSP, the major problem is how to avoid schedule interference and ensure that there is enough time available to setup and produce the lots selected to meet demands until the next production run. Soman et al. [38] proposed a model which allow products to be produced more than once in a cycle and also do not allow reducing production rate. Also, Brander and Segerstedt [8] modified the traditional cost function to include not only setup and inventory holding cost but also a time variable cost for operating the production facility.

In cyclic approaches, main alternatives are the common cycle (CC) introduced by Eilon [13], extended by Maxwell [28], the BP approach, developed by Bomberger [7] and the extended basic period, introduced by Elmaghraby [14]. In the former, each product is produced once in a CC and in the same cycle time, which is continuously repeated.

The BP approach, which was developed by Bomberger [7], assumes that production runs of all products shall be made in each basic period. In this approach there is a single BP, B, and each item is a replenishment cycle, T_i, is an integer multiple of B, namely, $T_i=k_i\times B$. In this manner, the problem objective will be finding a set of coefficients, $\{k_1, k_2, k_3,\ldots, k_N\}$, instead of finding a set of replenishment cycles $\{T_1, T_2, T_3,\ldots, T_N\}$. This approach ensures sequence feasibility by means of requiring a basic cycle long enough to accommodate production of all items once. In the literature, Cooke et al. [11] proposed a relatively simple MIP formulation for the ELS problem that creates a complete schedule, assuming a basic period value and production frequencies that have been predetermined.

The EBP expands upon the BP by allowing different cycle times for different products. Namely, by utilizing two consecutive fundamental cycles, but making them an integer multiple, k_i (for ith product), of some BP, long enough to accommodate a production run of all products. It has been found that the EBP approach is superior to the BP in respect of costs minimization, very significantly so in some cases. However, it tends to have much longer rotation cycle times. There is an extensive literature on this subject by referring to Elmaghraby [14] and Lopez and Kingsman [26].

In PoT policy, as pointed out earlier, it is assumed that $k_i=2^q$; $q\geq 0$ integer, for all k_i in the set of multipliers $K(B)$. A special case of the ELSP which assumes that the capacity of the production facility is defined by the annual available setup time is presented by Roundy [36]. Another special case of ELSP is studied by Jackson et al. [24] on the joint replenishment problem, where the capacity of the production facility is unlimited. Also, Federgruen and Zheng [16] use “unrestricted and stationary PoT policies” for multistage production and inventory systems.

In the lot sizing problems, if the deterioration of the items is ignored the demand may not be met. So it may cause additional costs due to shortage. A new inventory model in which products deteriorate at a constant rate and in which demand, production rates are allowed to vary with time has been introduced by Balkhi and Benkherouf [5]. In this model, an optimal production policy that minimizes the total relevant cost is established. Totally, most of the inventory models that considered the deteriorating factor are one-item models; for example, Misra [29], Elsayed and Teresi [15], Heng et al. [20], and Abad [1,2].

There are some classifications for deterioration. Ghare and Schrader [18] classified the inventory deteriorating into three categories: (1) direct spoilage, e.g., vegetable; (2) physical depletion, e.g., gasoline; and (3) deterioration in terms of loss of efficacy in inventory, e.g., medicine. Deterioration is classified by the life of the items of inventory by Nahmias [32] as follows: (1) fixed lifetime: independent of the deteriorating factors; (2) random lifetime: the probability distribution of the item could be an exponential distribution, etc. Also, Raafat [35] categorizes deterioration by the time-value of inventory: (1) utility constant: namely, its utility does not change significantly as time passes, e.g., liquid medicine; (2) utility increasing: its utility increases as time passes, like some alcoholic drink. (3) Utility decreasing: its utility decreases as time passes, e.g., fresh foods, etc. Totally, *time-dependent* deteriorating items are predicated to items that keep deteriorating in some probability distribution, e.g., electronic components, medicine, etc.

In lot sizing problems, if each item were the only item being produced, the answer is called the independent solution (IS). The set $T=\{T_1, T_2, T_3,\ldots, T_N\}$ is made up of the optimal T_i for each item, specified by T_i^*. If this solution is feasible, then the IS is the optimal solution. For problems with capacity utilization bigger than 0.25, unfortunately, this outcome is rare [19]. Usually, the IS is used as a lower bound for the conventional ELSP, although tighter lower bounds, which ensure enough capacity is available for setups, have been presented by Dobson [12].

Complicated problems are difficult to solve optimally. In many situations, a “good” solution acquired by a heuristic algorithm in reasonably short computational time is often desirable. Currently the most widely used heuristic techniques in combinatorial optimization are simulated annealing (SA), tabu search (TS), genetic algorithms (GAs), and ant colony optimization (ACOs) algorithms. Such evolutionary algorithms were suggested in the recent decades for solving optimization problems in different fields.

ICA is a new socio-politically motivated global search strategy that has recently been introduced for dealing with different optimization problem [4]. Nevertheless, its effectiveness, limitations, and applicability in various domains are currently being extensively investigated.

3 Problem description

In this section, we derive a mathematical model for the ELSP with deteriorating items and slack cost under PoT policy considering capacity constraint.

In the ELSP (with BP or EBP approach), the algorithm initially searches for an initial basic period B and its corresponding set of multipliers *K(B)*, and tries to obtain another basic period B' and its multipliers *K(B')* which improve the objective function value. Until obtaining no other basic period and its corresponding multipliers which improve the objective function value, the search continues. In intermediate steps of search, for sets of B and *K(B)*, one must either test its feasibility or obtain a feasible production schedule.

The conventional ELSP problem is determining a production schedule of i items (production cycle), where $i \in \{1, 2, 3, \ldots, n\}$ in a cyclical schedule [9]. If there is a time period T_i for each product that represents the time between consecutive production runs (batches or "lots") of item i, a cyclical schedule is achieved. This cyclical schedule is subject to the following assumptions related to the production facility:

1. Only one item i can be produced at a time; and number of all items is equal n.
2. Setting up for a certain item includes both a setup cost (a_i) and setup time (s_i).
3. Setup cost and setup time are determined merely by the product which is assigned on facility for production (sequence-independent).
4. Demand rate (d_i) and production rate (p_i) are known and constant for all items.
5. Holding costs (h_i) are determined by the quantity of the items held.
6. Total variable cost for an item equals average setup plus holding as well as slack cost over a specific period of time.
7. Production time for a batch of item i equals the sum of the processing time as well as setup time.
8. Shortage is allowed for all items, but is completely backlogged.
9. Each item deteriorates at an exponential rate θ_i and deterioration cost of per unit is equals ξ_i.
10. The deteriorated item cannot be repaired.
11. Each item has a due date (due$_i$) which must be delivered. Violating this assumption may cause a slack cost which is equal π_i. In other words, slacks are allowed for all items.

The solution of conventional ELSP consists of a set $T=\{T_1, T_2, T_3, \ldots, T_N\}$, such that each T_i is long enough to allow enough production of item i at the beginning of the cycle plus allow production of other items in the time left between the ends of production of item i and the start of the next cycle. If a set T is feasible and minimizes cost, it is optimal.

Two terms of objective function in conventional ELSP are: (1) setup cost denoted by a_i, incurs whenever the production facility sets up to produce the other items, and (2) inventory holding costs h_i. In addition to these two cost terms, we include the deterioration cost for the deteriorating items and slack cost for items violate the due date.

As pointed out earlier, to the best of the authors' knowledge, there is no other study directly surveying the conventional ELSP model that includes both the deterioration factor and slack cost under such aforementioned policies, however Yao and Huang [43] presented an ELSP model only consists of deteriorating factor. According to EBP approach under PoT policy with deteriorating items and slack cost, we present a mathematical model as follows:

$$\text{minimize TC}(\{k_i, B\}) = \sum_{i=1}^{n} \left\{ \frac{a_i}{k_i B} + \frac{1}{2} H_i k_i B + \lambda_i \pi_i \right\} \tag{1}$$

Subject to

$$\sum_{i=1}^{n} [(s_i + \beta_i(k_i, B))] w_{i\varphi(i,\tau)} \le B \quad ; \tau = 1, 2, 3, \ldots, K = lcm\{k_i\} = 2^{\max_i\{v_i\}} \tag{2}$$

$$\sum_{i=1}^{n} \frac{s_i + \beta_i(k_i, B)}{T_i} \le 1 \quad ; i = 1, 2, 3, \ldots, n \tag{3}$$

$$My_{ik} + t_i - t_k \ge p'_k \quad ; i = 1, 2, .., n \, ; \, i \le k \tag{4}$$

$$M(1 - y_{ik}) + t_k - t_i \ge p'_i \quad ; i = 1, 2, \ldots, n \, ; \, i \le k \tag{5}$$

$$k_i = 2^{v_i} \, ; \, v_i \in \{0, 1, 2, 3, \ldots\} \quad ; i = 1, 2, 3, \ldots, n \tag{6}$$

$$\sum_{\tau=1}^{k_i} w_{i\tau} = 1 \quad ; i = 1, 2, 3, \ldots, n \tag{7}$$

$$\varphi(i, \tau) = \begin{cases} \tau \bmod k_i & ; \mathit{if}\ \tau \ne \gamma k_i\,,\ \gamma \in N \\ k_i & ; \mathit{if}\ \tau = \gamma k_i\,,\ \gamma \in N \end{cases} \quad ; i = 1, 2, 3, \ldots, n \tag{8}$$

$$\begin{cases} w_{i\tau} = 1 \ ; \text{if product } i \text{ is produced in } \tau\text{th basic period} \\ w_{i\tau} = 0 \ ; \text{Otherwise} \quad ; \text{for all } i \text{ and } \tau \end{cases} \tag{9}$$

$$\begin{cases} \lambda_i = 1 & ; \text{if item } i \text{ has tardiness} \\ \lambda_i = 0 & ; \text{other wise} \end{cases} \quad ; i = 1, 2, 3, \ldots, n \tag{10}$$

$$H_i = d_i(\theta_i \xi_i + h_i) \quad ; i = 1, 2, 3, \ldots, n \tag{11}$$

$$\beta_i(k_i, B) = \frac{d_i}{p_i}\left(1 + \frac{k_i B \theta_i}{2}\right) k_i B \quad ; i = 1, 2, 3, \ldots, n \tag{12}$$

$$C_i = t_i + p_i' = t_i + (s_i + \beta_i(k_i, B)) \quad ; i = 1, 2, 3, \ldots, n \tag{13}$$

$$T_i' = \max\ \{0,\ C_i - \text{due}_i\} \quad ; i = 1, 2, 3, \ldots, n \tag{14}$$

Constraint (1) shows our cost function. Constraint (2) states that the total occupancy must be less than the length of basic period in each basic period τ (capacity constraint for a feasible production schedule). Constraint (3) ensures capacity feasibility or load feasibility which states the load never must exceeds the capacity of the facility. Constraints (4) and (5) together ensure that only one job can be processed at any instance in time. In other words, these precedence constraints state that start time of the job i is greater than or equal to the completion time of the job k, i.e., job i is latter in sequence rather than job k or vice versa. At any status, one of these constrains is redundant and the other one is active. Since by determining the sequence (using constraints (4) and (5)), the completion time of each item can be specified. In these constraints M is a large enough positive number and y_{ik}=0 or 1 and means if job i proceeds job k is the sequence, y_{ik}=1, else y_{ik}=0. Equation 6 shows the PoT policy. Equation 7 compels the production duration of job i must be scheduled among the first k_i basic periods. Constraint (8) identifies a basic period among the k_i basic periods belonging to product i. Actually, Eqs. 7 and 8 represent the starting basic periods of the production batches for all of the items. Equations 9, 10, and 11 are self-explained. Finally, Eq. 12 signifies the occupancy of each production batch for item i. Also, in Eq. 13 C_i and p'_i denote completion time and total processing time of job i (setup time plus occupancy of time of item i), respectively, and t_i is the start time of job i. Obviously, start time of each job depends on the previous jobs in sequence, i.e., it equals total spent time by prior jobs in sequence. Finally, T'_i in Eq. 14 is tardiness of job i.

The solution of our Slack-Deter-ELSP (EBP, PoT) problem consists of a set of multipliers $\{k_i\}$, value of the basic period (B) as well as a set of $\{\lambda_i\}$. A feasible production schedule for the obtained solution must be generated. To minimize the objective function, the ICA explores in the solution space of $\{k_i\}$. Since for a given set of multipliers $\{k_i\}$, the objective function is convex with respect to B, so we have $\frac{\partial TC(\{k_i\},B)}{\partial B} = 0$ to acquire the minimum of B value as follows:

$$B(\{k_i\}) = \sqrt{\frac{2\left(\sum_{i=1}^{n} \frac{a_i}{k_i}\right)}{\sum_{i=1}^{n} H_i k_i}} \tag{15}$$

Afterwards, we utilize Proc FT [43] heuristic for testing feasibility of ($\{k_i\}$, B). If there exists a feasible production schedule for the set ($\{k_i\}$, B), this schedule will be held as a nominee of the optimal solution, otherwise another schedule as primal schedule is produced to set a special value of B, that makes possible ($\{k_i\}$, B) to obtain a feasible production schedule with the minimum cost for the set $\{k_i\}$.

4 The proposed imperialist competitive algorithm

4.1 ICA in general

Imperialism is the strategy of expanding the power and rule of government beyond its own boundaries. There are several ways a country can be dominated by a powerful country; by means of direct rule or by less apparent instruments such as influence on culture, control of markets of raw materials, or other important commodity. Actually, ICA is a novel global search heuristic that uses imperialism and imperialistic competition process.

The ICA uses the socio-political process of imperialism and imperialistic competition as a source of inspiration [25]. The ICA initiates with an initial population, like most evolutionary algorithms. Each individual of the population is called a '*country*' equivalent '*chromosome*' in GA. Some of the most powerful countries are chosen to be the imperialist states and the other countries constitute the colonies of these imperialists. All the colonies of initial countries are partitioned among the mentioned imperialists based on their power. Equivalent of fitness value in the GA, the power of each country, is conversely proportional to its cost. An empire is constituted from the imperialist states with their colonies.

By constituting initial empires, each of their colonies begins progresses toward their related imperialist country. This is a simple kind of assimilation strategy which some of the imperialist states follow. Afterwards, the imperialistic competition starts among all the empires. Those empires which cannot succeed in this competition and are not capable to increase their power or at least prevent decreasing its power will be removed from the struggle. The imperialistic competition will slowly but surely result in an enhancement in the power of powerful empires and a decrease in the power of weaker ones.

The total power of an empire depends on both the power of the imperialist country and the power of its colonies. This fact is modeled by defining the total power of an empire as the power of imperialist country plus a percentage of mean power of its colonies [25].

Something which causes all the countries to converge to a state in which there exists only one empire in the world, is colonies movement toward their related imperialists along with struggle among empires and also the collapse mechanism. In such a case, all the other countries are colonies of that empire. In this ideal new world, colonies have the same position and power as the imperialist. Figure 1 shows the pseudo code for the proposed algorithm.

4.2 An ICA to Slack-Deter-ELSP (EBP-PoT)

In this section, we present the ICA for "Slack-Deter ELSP (EBP, PoT)" as a means of finding excellent production schedules of a cyclic nature.

4.3 Generating initial empires

The major purpose of optimization is to acquire an optimal solution. In our problem an array of variable values which must be optimized is formed. The term "country" in ICA is equivalent to "chromosome" in GA. Here, a country is a $1\times N_{\mathrm{var}}$ array which is defined by

$$\mathrm{country} = (g_1, g_2, g_3, \ldots, g_{N_{\mathrm{var}}})$$

Where each p_i is a variable which should be optimized. Each of these variables can be interpreted as a socio-political characteristic of a country, such as religion, culture, language, etc. From optimization perspective the solution with least cost value is the best one.

As in our problem, each multiplier must be represented as a specific section of a country, in order to encode the value of k_1, the first l_1 bits are employed to comply such a goal and the particular part of country from the (l_1+1)th bit to the (l_1+l_2)th bit represents the value of k_2 and so on. Country representation is shown in Fig. 2 to illustrate how k_is encode.

Actually, a country is composed of following matrix:

$$\mathrm{Country} = \begin{bmatrix} \mathrm{country}_1 \\ \mathrm{country}_2 \\ \mathrm{country}_3 \\ \\ \\ \mathrm{country}_{N_{\mathrm{pop}}} \end{bmatrix}$$

By evaluating the cost function f at variables (g_1, g_2, g_3,…, gN_{var}) the cost of a country is stated as follows:

$$\mathrm{Cost} = f(\mathrm{country}) = f(g_1, g_2, g_3, \ldots g_{N_{\mathrm{var}}})$$

Based on PoT policy, every multiplier k_i is a PoT integer, i.e., $k{=}2^{v}{}_i(v_i \in\{0,1,2,\ldots\})$. So, in order to encode in the country, we employ k_i by its integer value of power v_i. For better understanding, as an instance, if $l_i{=}2$, for presenting all the possible values of k_i, there exist $2^{li}{=}2^2{=}4$ possible value of v_i, i.e., {0, 1, 2, 3}, in which they correspond to {(0, 0), (0, 1), (1, 0), (1, 1)}, respectively, in binary coding. So, using the string {(1, 1)} is equivalent to $k_i{=}2^3$. To characterize all the possible values of k_i for each item i, an upper bound on the value of k_i and therefore on the value of integer-power v_i will need to include the country representation in ICA. In this context, we employ the value $\mathrm{TC}_i^{\mathrm{IS}}$, i.e., a lower bound on the average cost of item i using the independent solution (IS method), and $\mathrm{TC}^{\mathrm{IS}} = \sum_{i=1}^{n} \mathrm{TC}_i^{\mathrm{IS}}$ which denotes a lower bound on the objective cost function. Also $\mathrm{TC}^{\mathrm{RC}}$ is an upper bound based on rotational cycle (RC) approach [43].

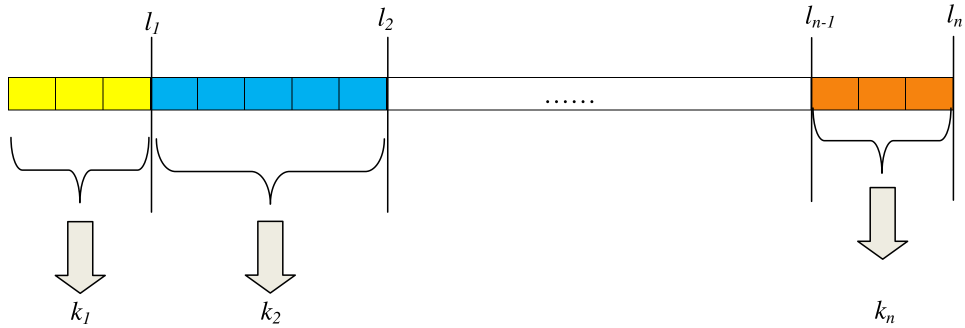
Yao and Huang [43] demonstrated that for a given value of B with deterioration items, an upper bound on v_i, indicated by $L^*_{\leq} B$, is:

$$v_i^{\mathrm{UB}}(B) = \left\lceil \log_2 \left(\frac{\mathrm{TC}^{\mathrm{RC}} - \sum_{j\neq i} \mathrm{TC}_j^{\mathrm{IS}} + \sqrt{\left(\mathrm{TC}^{\mathrm{RC}} - \sum_{j\neq i} \mathrm{TC}_j^{\mathrm{IS}}\right)^2 - 2d_i(\xi_i\theta_i + h_i)}}{B.d_i(\xi_i\theta_i + h_i)} \right) \right\rceil \quad (16)$$

Fig. 1 Pseudo code for the proposed imperialist competitive algorithm

1. ***Initialization***: Pick some random points on the function and initialize the empire
2. ***Assimilation***: Move the colonies toward their germane imperialist
3. Search all empires to ensure whether there's a colony in an empire which has lower cost than its imperialist. If so, swap the position of such a colony with its imperialist.
4. Compute the total cost of all empires (according to the imperialist and their colonies' power).
5. ***Imperialist competition***: Pick the weakest colony from the weakest empire and assign it to empire which has the most likelihood to posses it.
6. Eliminate the empire with no colonies.
7. If there's only one empire, stop, else go to Step 2.

Fig. 2 Country representation and coding scheme of multipliers for the proposed algorithm



In order to drive a bound on v_i, a lower bound on the value of B (B_{LB}) is needed. In this context, since the duration of B cannot be less than the total processing time of any item, it is impractical to have a feasible production schedule with $B<\max\left(s_i+\beta_i(s_i)\right)$. Consequently, we have:

$$B_{\mathrm{LB}}=\max_i\left(s_i+\beta_i(s_i)\right) \tag{17}$$

Now, an upper bound on$k_i^{'}s$, by$2^{v_i^{UB}(B_{\mathrm{LB}})}$is obtained where by replacing B_{LB} in the Eq. 16 we have:

$$v_i^{\mathrm{UB}}(B_{\mathrm{LB}})=\left[\log_2\left(\frac{\mathrm{TC}^{\mathrm{RC}}-\sum_{j\neq i}\mathrm{TC}_j^{\mathrm{IS}}+\sqrt{\left(\mathrm{TC}^{\mathrm{RC}}-\sum_{j\neq i}\mathrm{TC}_j^{\mathrm{IS}}\right)^2-2d_i(\xi_i\theta_i+h_i)}}{B_{\mathrm{LB}}.d_i(\xi_i\theta_i+h_i)}\right)\right] \tag{18}$$

Based on PoT policy, it is obvious that the lower bound on each v_i is $v_i^{\mathrm{LB}}=0$, because the lower bound on the value of k_i is 1. We encode the value of k_i by binary strings of integer powers, a mapping between each binary string and an integer must be established based on Yao and Huang [43]. Consequently, the total length of a country is $\sum_{i=1}^{n} l_i$ bits (Fig. 2).

In order to start the algorithm, first of all, we generate the initial population of size N_{pop}. We pick N_{imp} of the most powerful countries to constitute the empires. The left over N_{col} of the population will be the colonies each of which belongs to an empire (Fig. 3).

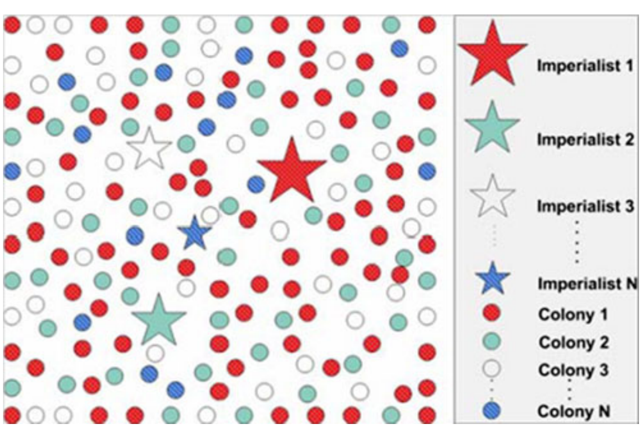


Fig. 3 Generating the initial empires: the more colonies an imperialist possess, the bigger is its relevant *star* mark

We split the colonies among imperialists based on their power, so as to constitute the initial empires. The initial number of colonies of an empire should be directly proportionate to its power. In order to partition the colonies among imperialists proportionally, the normalized cost of an imperialist is defined as follows:

$$\mathrm{NC}_\eta=\max_j\{\mathrm{dsb}_j\}-\mathrm{dsb}_\eta \tag{19}$$

Where dsb_η is the cost of ηth imperialist and NC_η is its normalized cost. The normalized power of each imperialist is acquired as follows:

$$\mathrm{pow}_j=\frac{\mathrm{NC}_j}{\sum_{j=1}^{N_{\mathrm{imp}}}\mathrm{NC}_j} \tag{20}$$

The normalized power of an imperialist, actually, is the portion of colonies that should be possessed by that imperialist. Then the initial number of colonies of an empire is

$$\mathrm{NOC}_j=\mathrm{round}\left\{\mathrm{pow}_j.N_{\mathrm{col}}\right\} \tag{21}$$

Where NOC_j is the initial number of colonies of jth empire and N_{col} is the number of all colonies. In order to partition the colonies, we pick NOC_n of the colonies at random for each imperialist and assign them to it. These colonies as well as the imperialist will constitute jth empire. Clearly, the larger empires have more colonies while weaker ones have less.

4.4 Movement of the colonies toward the imperialist (assimilation)

Countries of imperialists try to enhance their colonies. This fact has been modeled by moving all the colonies toward the imperialist. The colony will approach to the imperialist along different socio-political axis such as culture, language, etc. In other words, imperialist states change socio-political characteristics of colonies in such a way that they become similar to them (increase their power). All the colonies will be fully assimilated into the imperialist by keeping on this action.

Through this movement some parts of the structure of a colony will be similar to the structure of the empire. The assimilating operator is shown with an example in Figs. 4 and 5.

- Select randomly one cell in imperialist array (for example cell 6, number 3).
- Find this number in the array of the colony and shift this number to reach to the same position as in imperialist array (cell 2, number 3).
- Put the right hand side number in the array of the imperialist (in this example it is 5) at the right hand side of the shifted cell in the array of the colony (swap numbers 5 and 8).

4.5 Exchanging positions of the imperialist and a colony

Meanwhile moving toward the imperialist, a colony may get to a situation with lower cost than the imperialist. In such a status, the position of the imperialist and the colony are change. Thereafter, the algorithm will keep on by the imperialist in the new position and the colonies will be assimilated by the imperialist in its new position.

Figure 6a depicts the position exchange between a colony and the imperialist. The best colony of the empire is shown in a darker color in this figure so that its cost is lower than the imperialist. Figure 6b shows the whole empire after exchanging the position of the imperialist and that colony.

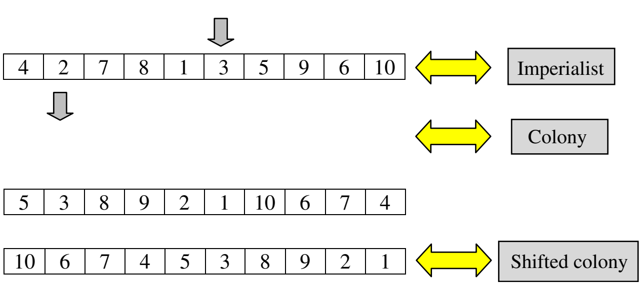


Fig. 4 Shifting part of the assimilating operator

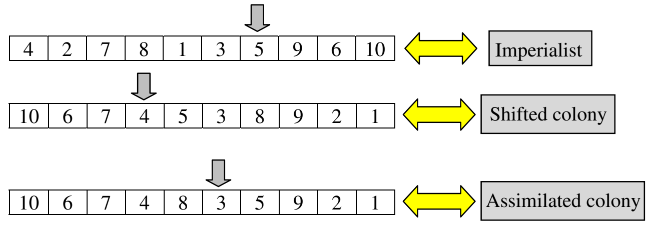


Fig. 5 Swapping part of the assimilating operator

4.6 Total power of an empire

One of the factors which affect on the total power of an empire is the power of imperialist country (actually, the most influential factor). However, piddling, but the power of the colonies of an empire has an effect on the total power of that empire. This reality is modeled by defining the total cost of an empire as follows:

$$\mathrm{TC}_j = \mathrm{Cost}\left(\mathrm{imperialist}_j\right) + \gamma \ \mathrm{mean}\left\{\mathrm{cost}\left(\mathrm{colonies\ of\ empire}_j\right)\right\} \tag{22}$$

Where TC_j is the total cost of the jth empire and γ is a positive small number. If values of γ are little, the total power of the empire will be determined by approximately only the imperialist and vice versa.

4.7 Imperialistic competition

As pointed out earlier, all empires try to possess the other colonies of the empires to control them. By keeping on the imperialistic competition, the power of weaker empires will decrease and the power of more powerful ones will reinforce. In this context, the imperialistic competition is modeled by choosing one of the weakest colonies of the weakest empire and making a competition among all empires to possess this colony. Each of the empires (based on its total power) will have a likelihood of taking possession of the mentioned colonies. It is notable that the most powerful empires will not absolutely possess these colonies, but merely they are more likely to possess them. Figure 7 depicts how imperialist completion modeled.

A colony of the weakest empire is chosen as the first step to initiate the competition and the possession probability of each empire (P_{pos}) is then found which is proportionate to the total power of the empire. The normalized total cost of an empire can be acquired by Eq. 23:

$$\mathrm{NTC}_\eta = \max_j\left\{\mathrm{TC}_j\right\} - \mathrm{TC}_\eta \tag{23}$$

Where TC_η and NTC_η are the total cost and the normalized total cost of ηth empire, respectively. Having the

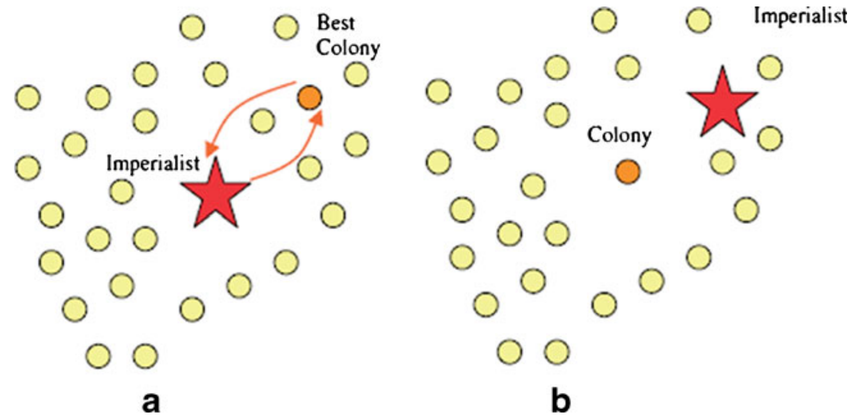


Fig. 6 **a** Exchanging the positions of a colony and the imperialist; **b** the entire empire after position exchange

normalized total cost, the possession probability of ηth empire will be known as follows:

$$P_{\mathrm{pos}_\eta} = \frac{\mathrm{NTC}_\eta}{\sum_{j=1}^{N_{\mathrm{imp}}} \mathrm{NTC}_j} \tag{24}$$

Choosing an empire is analogous to the roulette wheel process which is used in selecting parents in GA, but since in this method calculation of the cumulative distribution function is not needed, i.e., the selection is based on only the values of probabilities of selection. Consequently, this method is much faster than the conventional roulette wheel. Based on aforementioned explanations, the process of choosing the empires can substitute the roulette wheel in GA and therefore causes an increase in its computational speed.

In order to partition the mentioned colonies among empires based on the possession probability of them, vector ***P*** is formed as follows:

$$P = \left[p_{\mathrm{pos}_1}, p_{\mathrm{pos}_2}, \dots, p_{\mathrm{pos}_{N_{\mathrm{imp}}}}\right]$$

Afterward, we create a vector ***R*** with the same size as ***P*** whose elements are uniformly distributed random numbers as follows:

$$R = \left[r_1, r_2, \dots, r_{N_{\mathrm{imp}}}\right] \quad ; r_1, r_2, \dots, r_{N_{\mathrm{imp}}} \in U(0, 1)$$

Vector ***E*** can be acquired by subtracting ***R*** from ***P***:

$$E = P - R = \left[E_1, E_2, \dots E_{N_{\mathrm{imp}}}\right] = \left[p_{\mathrm{pos}_1} - r_1, p_{\mathrm{pos}_2} - r_2, \dots, p_{\mathrm{pos}_{N_{\mathrm{imp}}}} - r_{N_{\mathrm{imp}}}\right]$$

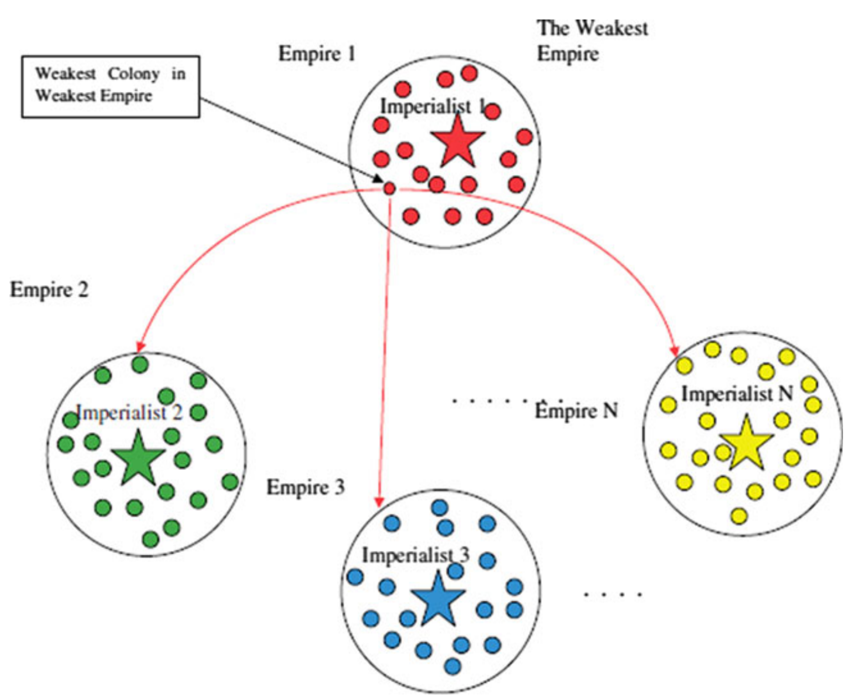


Fig. 7 The more powerful an empire is, the more likely it will possess the weakest colony of the weakest empire (Imperialistic competition)

```
t ← 0
Initialize population (t)
Evaluate population (t)
  While (stopping criteria is met)
t ← t +1
Selection: apply LRN beside roulette wheel mechanism
Elitism: 20% individuals with the best fitness values
        Mutation: mr (t + 1) = mr (t) × 1.01
        Uniform crossover: cr (t + 1) = cr (t) × 0.999
Evaluate population (t)
```

Fig. 8 Pseudo code for the GA

Referring to vector $\boldsymbol{E}$, the mentioned colony (colonies) will hand to an empire whose relevant index in $\boldsymbol{E}$ is maximum.

4.8 Elimination of the powerless empires

Those empires which are powerless will collapse in the imperialistic competition and their colonies will be partitioned among other empires. Different criteria can be introduced so as to consider a powerless empire in collapse mechanism modeling, such as “time limitation” or “maximum number of iterations”. When an empire loses all of its colonies, in fact, this empire is collapsed and consequently, will be eliminated. This is the mechanism which we use in our problem.

4.9 Convergence and stop criterion

By keeping on the algorithm and spending the time, all of the empires will collapse except the most powerful one and all the colonies will be subjected to this empire. In such an ideal new world, all the colonies will have the same positions and same costs and they will be controlled by an imperialist with the same position and cost as themselves. In such a world, there is no difference not only among colonies, but also between colonies and imperialist.

5 GA for Slack-Deter-ELSP (EBP, PoT)

In this section, we present the GA for Slack-Deter-ELSP (EBP, PoT) problem as a common tool for obtaining the near-optimal solution. As mentioned in the Section 4.2, an array of variable values which is called “chromosome” is defined by a $1\times N_{\text{var}}$ matrix. This array is defined by

$$\text{chromosome} = \left(g_1, g_2, g_3, \ldots, g_{N_{\text{var}}}\right)$$

Since structure of a “chromosome” is like to a “country”, all components of a chromosome are the same as a “country”. Now, since there may exist problems associated with fitness values when solving minimization problems, we use fitness normalization in our GA. Fitness normalization is a process of converting row fitness values to ones that behave better [23] and give high probability for selecting good solutions in new generations, while maintaining some chance of survival to poor solutions [6]. The term selection pressure (sp) represents the ratio of the probability of selecting the best individual to the average probability of selecting all individuals [34]. We use linear ranking normalization (LRN) in our problem. In LRN, all of the individuals in a population are ranked and stored on a temporary list. Ranking of the individuals is carried out according to their fitness. The size of a population is specified as sp and the index of an individual within the temporary list as i_{temp}. Then, the best-fit individual stores the first portion of the list and has the highest rank in the list i_{temp}=sp. The sp takes values in the range of [1.0, 2.0]. This value in this problem is regarded equal 1.5. By Yao and Huang [43], the normalized fitness values of individual i_{temp} (within the temporary list) are calculated as follows:

$$\text{Eval}_{i_{\text{temp}}} = 2 - \text{sp} + \frac{2(\text{sp}-1)\left(i_{\text{temp}}-1\right)}{\text{ps}-1} \tag{25}$$

For selection mechanism, we use roulette wheel strategy for selection of individuals in reproduction. The reproduction

Table 1 Factor levels for small, medium, and large problems

Factor	Symbol	Levels	Type of problems
γ	*A*	3	Small: *A*(1)=0.002, *A*(2)=0.005, *A*(3)=0.01
			Medium: *A*(1)=0.005, *A*(2)=0.01, *A*(3)=0.02
			Large: *A*(1)=0.02, *A*(2)=0.05, *A*(3)=0.1
N_{imp}	*B*	3	Small: *B*(1)=3, *B*(2)=4, *B*(3)=6
			Medium: *B*(1)=3, *B*(2)=5, *B*(3)=8
			Large: *B*(1)=8, *B*(2)=12, *B*(3)=16
($ngen$,N_{country})	C	3	Small: *C*(1)=(240, 100), *C*(2)=(200, 120), *C*(3)=(120, 200)
			Medium: *C*(1)=(200, 120), *C*(2)=(120, 200), *C*(3)=(100, 240)
			Large: *C*(1)=(240, 100), *C*(2)=(200, 120), *C*(3)=(120, 200)

Table 2 Taguchi orthogonal array design

No	γ	N_{imp}	($ngen$, N_{country})
1	1	1	1
2	1	2	2
3	1	3	3
4	2	1	2
5	2	2	3
6	2	3	1
7	3	1	3
8	3	2	1
9	3	3	2

probability of each individual is proportional to its normalized fitness as expressed in Eq. 26 as follows:

$$P_{i_{\text{temp}}} = \frac{\text{Eval}_{i_{\text{temp}}}}{\sum_{i_{\text{temp}}=1}^{\text{ps}} \text{Eval}_{i_{\text{temp}}}} \quad (26)$$

Uniform crossover strategy is applied to this problem because it is assumed to reduce the bias associated with the length of the binary representation used and the particular coding for a given parameter set [34]. The mutation operator randomly selects ones among the genes of all individuals in the population with a fixed mutation rate (mr). The population size is recommended to be set as PS=10n (n=5, 10 and 15). The crossover rate (cr) and the mutation rate vary linearly during the evolutionary process. We set the cr at a higher level (0.9) in the start of the evolution while the mr is lower (0.05), in order that our GA can take advantage of the characteristics of the individual. Through the evolutionary process, the crossover rate for each generation decreases by 0.001 and the mutation rate increases by 0.01 after 100 generations. The variation of cr and mr stop as they reach a specified level, i.e., cr=0.2 and mr=0.2. Using such policy our GA is capable still to explore new area in the search space and increase the population diversity, since at the end of evolution the individuals become similar to each other (because of cr decreases while mr increases). In our paper, one of the "time limitation" or "50 nonimprovement iterations" is determined as stopping criteria. Our GA procedure is shown in Fig. 8 as follows:

Table 3 Problems data sets

Input variables	Distribution
Demand rate (d_i)	~ DU[2,000, 60,000]
Produce rate (p_i)	~ DU[5,000, 125,000]
Holding cost (h_i)	~ DU[5, 120]
Setup cost (a_i)	~ DU[60, 600]
Setup time (s_i)	~ DU[5, 15]
Deterioration cost (ξ_i)	~ DU[10, 110]
Deterioration factor (θ_i)	~ U[0.3, 3]
Due date (due_i)	~ DU[30, 85]
Slack cost (π_i)	~ DU[80, 500]

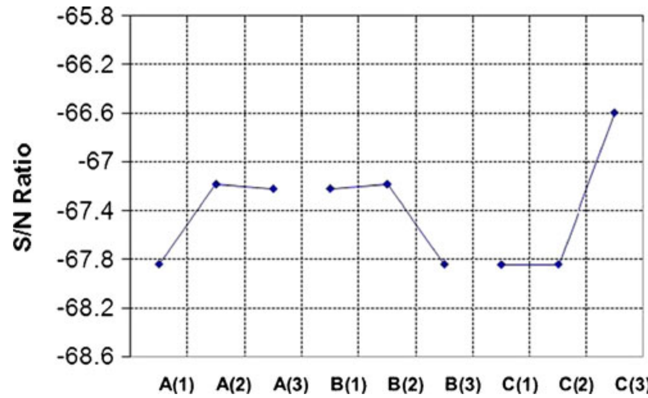


Fig. 9 Main effect plot for S/N ratios for small problems

6 Generation of feasible production schedule

At first, we need to obtain a production schedule of each item by assigning the production lots of all the items. Furthermore, we have to determine the set of variables $\{w_{it}\}$.

In the literature, Yao et al. [42] showed being NP-completeness of the problem of generating a feasible production schedule for a given set of ($\{k_i\}$, B) for the conventional ELSP (EBP, PoT) model. Showing the complexity of NP-completeness of the Deter-ELSP (EBP, PoT) model is very easy. So being NP-completeness of Slack-Deter-ELSP (EBP-PoT) is evident. For obtaining a feasibility testing procedure, we use Proc FT [43] with some changes. Suppose G signify a candidate schedule and $L(G)$ be the maximal load secured by G. Presume that a set of multipliers $\{k_i\}$ and B is given. The corresponding occupancy times β_i (k_i, B)} evidently will be determined. Use a random production schedule to acquire an initial schedule of production G, and calculate $L(G)$. Regard L^* as the best load secured up to

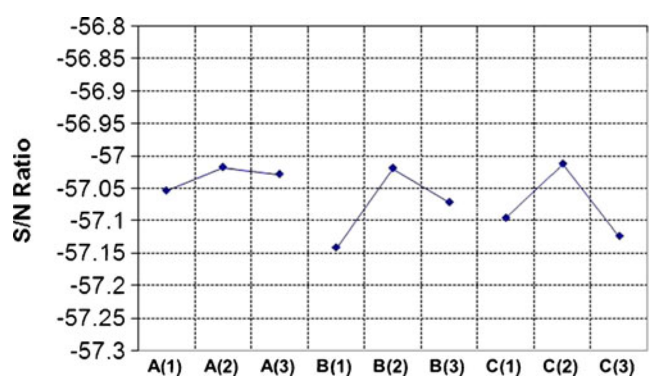


Fig. 10 Main effect plot for S/N ratios for medium problems

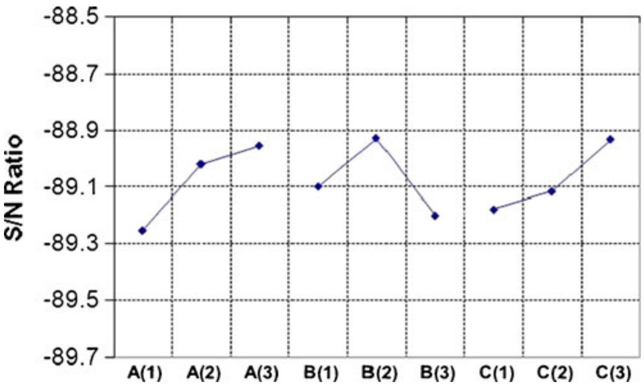


Fig. 11 Main effect plot for S/N ratios for large problems

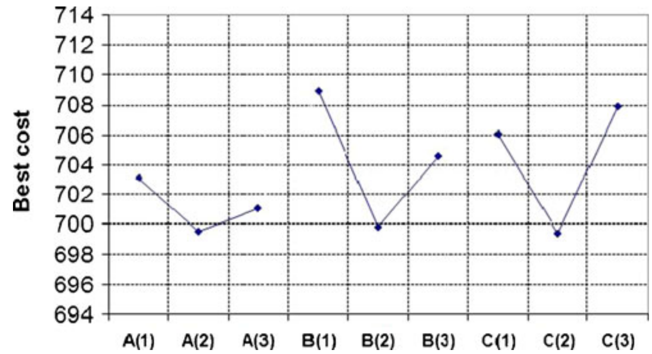


Fig. 13 The mean best cost plot for each level of the factors (medium problems)

now, and G^* its corresponding production schedule. (If G is the first production schedule then set $L^*=L(G)$ and $G^*=G$). When $L^*\leq B$, clearly the recommended assignment is a feasible production schedule. Here, indicator μ is defined as follows; if a feasible production schedule is acquired in Proc FT, $\mu=1$; otherwise, $\mu=0$. After generation a random production schedule as a random solution, if $\mu=0$, i.e., no feasible production schedule is attained, use the "Schedule Smoothing Procedure" [43] to improve the maximal load secured by candidate schedule, L^* until $\mu=1$ or L^* can no longer be improved. If L^* has not been improved for a constant consecutive iterations (*maxit*), Stop. Or else, select a subset of items for re-optimization randomly; fix the schedule for the rest of the items, and return to start another local search iteration. The constant *maxit* is arbitrary criterion for stopping the proposed heuristic algorithm which is defined by user opinion.

7 Experimental design

Suitable design of the operators and parameters has an important impact on the efficiency of the ICA. In order to calibrate the algorithms, there are several ways to statistically design the experimental investigation, but the most frequently used and exhaustive approach is a full factorial experiment [30]. This approach cannot be always effective since it becomes increasingly difficult to carry out investigations when the number of factors becomes considerably large. In order to reduce the number of required tests, a fractional factorial experiment (FFE) was developed [10]. FFE permits just a segment of the total possible combinations to approximate the main effect of the factors and some of their interactions. Taguchi [39] developed a family of FFE matrices that ultimately lessens the number of experiments, but still provides adequate information. In the Taguchi method, orthogonal arrays are employed to study a large number of decision variables with a small number of experiments.

Taguchi splits the factors into two main groups: controllable and noise factors. Noise factors are those over which we cannot directly control them. Since removal of the noise factors is impractical and often impossible, the Taguchi method looks for to minimize the effect of noise and to determine the optimal level of the important controllable factors based on the concept of robustness [40]. Besides determining the optimal levels, Taguchi establishes the relative importance of individual factors in terms of their main effects on the objective function.

Taguchi created a transformation of the repetition data to another value which is the measure of variation. The

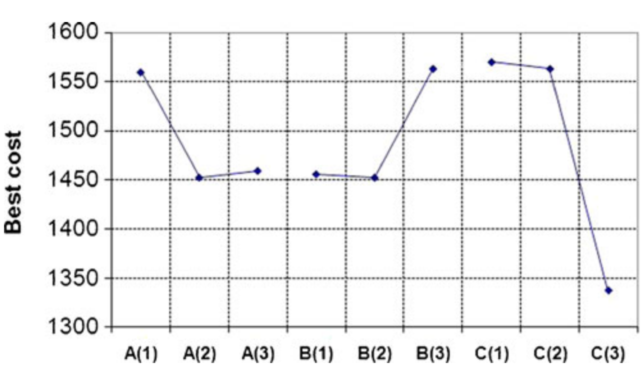


Fig. 12 The mean best cost plot for each level of the factors (small problems)

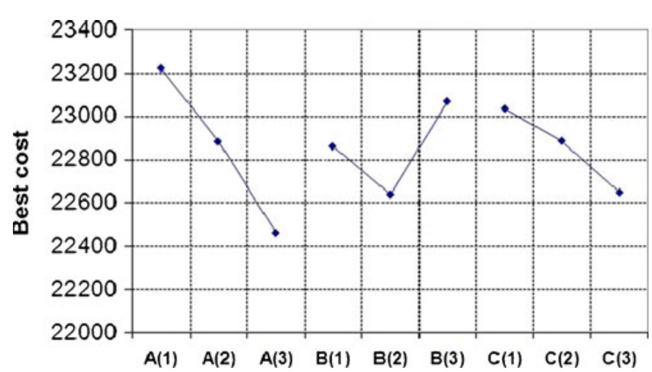


Fig.14 The mean best cost plot for each level of the factors (large problems)

Table 4 Average results for the algorithms grouped by size

Problem size	GA			ICA		
	Best cost	CPU time	Best time	Best cost	CPU time	Best time
Small	1,275.35	539.33	262.40	942.08	179.69	4.24
Medium	3,417.32	912.11	601.42	707.64	355.53	14.64
Large	22,280.82	1,153.73	585.44	21,791.34	515.33	65.43

transformation is the signal-to-noise (S/N) ratio, which explains why this type of parameter design is called a robust design [3,33]. The term "signal" indicates the desirable value (response variable) and "noise" signifies the undesirable value (standard deviation). Therefore, the S/N ratio specifies the amount of variation present in the response variable. Here, maximization of the signal-to-noise ratio is addressed (i.e., is the goal).

Taguchi categorizes objective functions into three groups: the smaller-the-better type, the larger-the-better type, and the nominal-is-best type. Since almost all objective functions in scheduling are classified in the smaller-the-better type, the corresponding S/N ratio [33] is

$$\frac{S}{N}\ \text{ratio} = -10\ \log\ (\text{objective function})^2 \tag{27}$$

7.1 Date generation and setting

An experiment was conducted to test the performance of the ICA. In this study, the control factors are: multiplier of power of colonies effectiveness in ICA (γ), number of imperialist (N_{imp}) and total number of countries (N_{country}) with number of generations (ngen) together as one factor (ngen, N_{country}). As universalization, in this paper we study three different categories of problems, i.e., small problems (five items), medium problems (10 items), and large problems (15 items). For each instance, because of different size of problems, the factors level differs. Tables 1, 2, and 3 show different levels of these factors for all small, medium, and large problems, respectively.

Since we have three three-level factors, the total number of trials required for each category would be a full combination of 27 (3^3) trials, rather than nine trials by the orthogonal array $L_9(3^3)$.

7.2 ICA parameters tuning based on Taguchi method

It is known that the different levels of the parameters clearly affect the quality of the solutions obtained by ICA. A number of different ICA can be obtained with the different combinations of the parameters. We have applied parameters tuning for aforementioned factors, namely, γ, N_{imp}, and (ngen, N_{country}). In Table 2, the Taguchi orthogonal array design used for this study is shown which is extracted from Minitab software.

In order to conduct the experiments, we implemented ICA in MATLAB 7.5.0 run on a PC with a 1.83 GHz Intel Core 2 Duo processor and 2 GB RAM memory. By acquiring the results of the Taguchi experiment for all the trials, all objective functions are individually transformed into S/N ratios. Figures 9, 10, and 11 show the average S/N ratio obtained at each level for small, medium, and large problems, respectively. We use three test examples for small,

Table 5 The best computational results of sample problems

Type	No.	$\sum_{i=1}^{n} \rho_i$	TC* ($)	B (days)	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_9	k_{10}	k_{11}	k_{12}	k_{13}	k_{14}	k_{15}
Small	1	0.84	4,400.6	31.5	4	4	4	2	4	–	–	–	–	–	–	–	–	–	–
	2	2.44	175.1	21.0	8	8	4	2	64	–	–	–	–	–	–	–	–	–	–
	3	3.14	250.1	15.8	16	16	16	16	4	–	–	–	–	–	–	–	–	–	–
Medium	1	6.33	820.1	4.5	64	16	4	8	8	16	32	8	8	16	–	–	–	–	–
	2	2.86	722.6	20.3	16	32	8	8	8	32	16	16	4	8	–	–	–	–	–
	3	6.33	720.2	5.1	4	16	8	4	4	16	32	64	4	4	–	–	–	–	–
	4	6.69	520.1	6.2	8	8	16	8	4	8	16	8	8	16	–	–	–	–	–
Large	1	2.52	35,000.8	33.2	8	4	8	4	4	16	16	32	8	32	16	8	16	8	32
	2	7.33	1,350.1	26.4	8	8	4	4	16	8	8	16	16	16	8	8	8	8	4
	3	9.41	24,001.7	26.9	8	4	4	4	4	8	8	8	16	32	16	64	16	32	8
	4	7.32	24,300.6	12.3	16	8	16	16	32	16	16	16	16	8	64	16	16	32	32
	5	8.53	28,000.8	20.9	8	4	4	16	16	8	32	16	16	16	8	32	8	32	8

four for medium, and five for large problems. Each problem has run four times for nine scenarios.

As indicated in Fig. 9, the optimal level of the factors *A*, *B*, is almost *A*(2) and *B*(2), respectively. However, determining the optimal level for all factors necessitates more investigation. In order to do so, we analyze the results of the experiment using a different measure, objective function. The results for each level are shown in Fig. 12. This analysis strongly supports our decision with respect to the optimal level for factors *A*, *B*, and *C*. It finally turns out that *A*(2) and *B*(2) are the preferable levels for factors *A* and *B*, respectively. Also, for factor *C*, the last parameter, i.e., *C*(3) is preferable.

In Fig. 10, the optimal level of the factors *A*, *B*, and *C* clearly becomes *A*(2), *B*(2), and *C*(2), respectively. But for factor *A* we use objective function as a different measure. Figure 13 demonstrates the mean best cost plot for different levels of factors for medium problems. The obtained result powerfully confirms our assessment regarding the best (optimal) level for factors *A*, namely *A* (2) is optimum.

In Fig. 11, result illustrates that for factor *B*, the optimal level is *B*(2). For factor *A* there is not a significant difference between levels 2 and 3; however, the third level is preferable superficially. Also for the factor *C*, the third level is optimal. But we utilize again objective function as a different measure. Figure 14 shows that the optimal levels for factors *A*, *B*, and *C* is *A*(3), *B*(2), and *C*(3), respectively.

8 Experimental results

In this section we are going to compare the proposed ICA with the GA, both for the Slack-Deter-ELSP (EBP, PoT). The heuristics were implemented in MATLAB 7.5.0 and run on a PC with a 1.83-GHz Intel Core 2 Duo processor and 2 GB RAM memory.

As pointed out earlier, we generate three small, four medium, and five large problems randomly where input data sets are illustrated in Table 3.

Each instance in all categories has run four times on each algorithm so as to ensure the stable respond of the algorithms. Since we have three categories of problems, we compare the results for all small, medium, and large problems on GA and proposed ICA separately. Table 4 shows average results for the best obtained parameters. As can be seen, the obtained results demonstrate high performance of ICA in respect of GA, i.e., the ICA outperforms GA in all instances in the considered characteristics. The best computational results of sample problems in all categories for Slack-Deter-ELSP (EBP, PoT) problem are shown in Table 5.

9 Conclusions and future work

In this paper, an imperialist competitive algorithm for solving multi-criteria economic lot scheduling problem with deterioration items and slack costs regarding capacity constraint using the extended basic period approach under the PoT policy is proposed.

To solve the considered problem, we employ ICA which is equipped with a feasibility testing procedure. We exhaustively explore the different parameters of the imperialist competitive algorithm by means of Taguchi method. In order to evaluate the effectiveness and robustness of the proposed ICA, we carry out a comparison between our ICA and GA. In this context, we use three different categories of test problems in small, medium, and large sizes and each instance has run four times so as to make certain the steady respond of the algorithms. The computational results demonstrate that our proposed ICA is an efficient solution approach for solving Slack-Deter-ELSP (EBP, PoT) and outperforms the GA.

As a direction for future research, it would be interesting to develop single objective ELSP into multi-objective which is closer to the real world conditions. Another direction for future research is developing discrete version of ICA for solving such difficult discrete problems. One more opportunity for research is how to birth the empires in the ICA.

References

1. Abad PL (1996) Optimal pricing and lot-sizing under conditions of perish ability and partial backordering. Manag Sci 42:1093–1104
2. Abad PL (2000) Optimal lot size for a perishable good under conditions of finite production and partial backordering and lost sale. Comput Ind Eng 38:457–465
3. Al-Aomar R (2006) Incorporating robustness into genetic algorithm search of stochastic simulation outputs. Simulat Model Pract Theor 14:201–223
4. Atashpaz E, Lucas C (2007) Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialist Competition. 2007 IEEE Congress on Evolutionary Computation (CEC 2007).
5. Balkhi Z, Benkherouf L (1996) A production lot size inventory model for deteriorating items and arbitrary production and demand rates. Eur J Oper Res 92:302–309
6. Boesel J, Nelson BL, Ishii N (2003) A framework for simulation-optimization software. IIE Transactions 35:221–229
7. Bomberger E (1966) A dynamic programming approach to a lot scheduling problem. Manag Sci 12:778–84
8. Brander P, Segerstedt A (2009) Economic lot scheduling problems incorporating a cost of using the production facility. Int J Prod Res 47:3611–3624
9. Chatfield DC (2007) The economic lot scheduling problem: a pure genetic search approach. Comput Oper Res 34:2865–2881
10. Cochran WG, Cox GM (1992) Experimental designs, 2nd edn. Wiley, New York
11. Cooke DL, Rohleder TR, Silver ED (2004) Finding effective schedules for the economic lot scheduling problem: a simple mixed integer programming approach. Int J Prod Res 42:21–36

12. Dobson G (1987) The ELSP: achieving feasibility using time-varying lot sizes. Oper Res 35(5):764–71
13. Eilon S (1958) Economic batch size determination for multiproduct scheduling. Oper Res Q 10:217–277
14. Elmaghraby SE (1978) The economic lot scheduling problem (ELSP): review and extension. Manag Sci 24:587–597
15. Elsayed EA, Teresi C (1983) Analysis of inventory system with deteriorating items. Int J Prod Res 21:449–460
16. Federgruen A, Zheng YS (1993) Optimal power-of-two replenishment strategies in capacitated general production distribution networks. Manag Sci 39:710–727
17. Gallego G, Joneja D (1994) Economic lot scheduling problem with raw material considerations. Oper Res 42(1):92–101
18. Ghare PM, Schrader GF (1963) A model for an exponentially decaying inventory. J Ind Eng 14:238–243
19. Goyal SK (1975) Scheduling a multi-product single machine system—a new approach. Int J Prod Res 13:487–93
20. Heng KJ, Labban J, Linn RJ (1991) An order-level lot-size inventory model for deteriorating items with finite replenishment rate. Comput Ind Eng 20:187–197
21. Heydari M, Torabi SA (2008) The economic lot scheduling problem in flow lines with sequence-dependent setups. World Academy of Science Eng Tech 47:2008
22. Hsu WL (1983) On the general feasibility of scheduling lot sizes of several products on one machine. Manag Sci 29:93–105
23. Hunter A (1998) Crossing over genetic algorithms: the Sugal generalized GA. Journal of Heuristics 4:179–192
24. Jackson PL, Maxwell WL, Muckstadt JA (1985) The joint replenishment problem with a power-of-two restriction. IIE Trans 17:25–32
25. Khabbazi A, Atashpaz E, Lucas C (2009) Imperialist competitive algorithm for minimum bit error rate beamforming. Int J Bio-Inspired Comput 1:125–133
26. Lopez MA, Kingsman BG (1991) The economic lot scheduling problem: theory and practice. Int J Prod Econ 23:147–164
27. Luss H, Rosenwein MB (1990) A lot-sizing model for just-in-time manufacturing. J Oper Res Soc 41(3):201–209
28. Maxwell WL (1964) The scheduling of economic lot sizes. Nav Res Logist Q 11:89–124
29. Misra RB (1975) Optimal production lot size model for a system with deteriorating items. Int J Prod Res 13:495–505
30. Montgomery DC (2000) Design and analysis of experiments, 5th edn. Wiley, New York
31. Moon IK, Cha BC, Bae HC (2006) Hybrid genetic algorithm for group technology economic lot scheduling problem. Int J Prod Res 44:4551–4568
32. Nahmias S (1978) Perishable inventory theory: a review. Oper Res 30:680–708
33. Phadke MS (1989) Quality engineering using robust design. Prentice-Hall, Engelwood Cliffs
34. Pohlheim H (2001) Evolutionary algorithms: principles, methods, and algorithms [online]. Available from: http://www.geatbx.com/index.html, [accessed on 6 May 2004].
35. Raafat F (1991) Survey of literature on continuously deteriorating inventory models. J Oper Res Soc 42:27–37
36. Roundy R (1984) Rounding off to powers of two in continuous relaxations of capacitated lot sizing problems. Manag Sci 35:1433–1442
37. Soman CA, Van Donk DP, Gaalman G (2003) Combined make-to-order and make-to-stock in a food production system. To appear in International Journal of Production Economics, http://dx.doi.org/10.1016/S0925-5273(02)00376-6.
38. Soman CA, Van Donk DP, Gaalman G (2004) A basic period approach to the economic lot scheduling problem with shelf life considerations. Int J Prod Res 42:1677–1689
39. Taguchi G (1986) Introduction to quality engineering. Asian Productivity, White Plains
40. Tsai JT, Ho WH, Liu TK, Chou JH (2007) Improved immune algorithm for global numerical optimisation and job-shop scheduling problems. Appl Math Comput 194:406–424
41. Yao MJ, Elmaghraby SE (2001) The economic lot scheduling problem under power-of-two policy. Comput Math Appl (CMWA) 41:1379–1393
42. Yao MJ, Elmaghraby SE, Chen IC (2003) On the feasibility testing of the economic lot scheduling problem using the extended basic period approach. J Chin Inst Ind Eng 20:435–448
43. Yao MJ, Huang JX (2005) Solving the economic lot scheduling problem with deteriorating items using genetic algorithms. J Food Eng 70:309–322