

Contents lists available at [SciVerse ScienceDirect](http://SciVerse.ScienceDirect.com)

Applied Mathematical Modelling

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

A particle swarm algorithm for inspection optimization in serial multi-stage processes

Ali Azadeh*, Mohamad Sadegh Sangari, Alireza Shamekhi Amiri

Department of Industrial Engineering, University College of Engineering, University of Tehran, Iran

ARTICLE INFO

Article history:

Received 15 September 2010

Received in revised form 27 August 2011

Accepted 1 September 2011

Available online 8 September 2011

Keywords:

Inspection policy

Particle swarm optimization (PSO)

Serial multi-stage process

Total inspection cost

Sample size

ABSTRACT

Implementing efficient inspection policies is much important for the organizations to reduce quality related costs. In this paper, a particle swarm optimization (PSO) algorithm is proposed to determine the optimal inspection policy in serial multi-stage processes. The policy consists of three decision parameters to be optimized; i.e. the stages in which inspection occurs, tolerance of inspection, and size of sample to inspect. Total inspection cost is adopted as the performance measure of the algorithm. A numerical example is investigated in two phases, i.e. fixed sample size and sample size as a decision parameter, to ensure the practicality and validity of the proposed PSO algorithm. It is shown that PSO gives better results in comparison with two other algorithms proposed by earlier works.

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

The organizations nowadays have to reduce costs through managing different aspects of their business operations to continuously improve products, services, and processes. They must satisfy the customers' needs and expectations to survive in the severe competitive environment. Since quality of services and products is one of critical factors that significantly affects customers' satisfaction, developing efficient quality systems is then very important for organizations.

According to Emmons and Rabinowitz [1], implementing a quality system is expensive and requires valuable resources of the organization. Moreover, the products and production systems become more and more complex with a larger set of failure possibilities. As the organizations invest large amounts in such systems, implementing an efficient inspection strategy is of much importance to reduce quality related costs. Therefore, the quality economics is a major issue for the organizations.

Many organizations establish inspection systems as a tool to achieve quality. However inspection is an inferior way of dealing with quality problems, but the benefits of quality improvement are superior to any inspection scheme for many cases [2]. Identifying an efficient inspection policy has economic relevance, as adopting different inspection policies will result in different costs. The inspection policy may affect the production process in different stages, thus inspection only in the last stage may cause to non-conformance products, waste resources, and incur penalty costs because of customer dissatisfaction, losing market share, etc. Therefore, a cost trade-off is necessary in selecting the optimal inspection policy [2]. On the other hand, subjecting a larger product fraction to inspection, or tightening the acceptance limits, will normally lead to a higher product quality, but will result in higher costs of inspection, scrap, and rework [3]. Therefore, establishing an efficient economic inspection policy is desired to balance these effects. The policy should ensure the required output quality while minimizing total inspection cost. The total inspection cost comprises the cost of all inspected units and the cost incurred by defective units detected at any stage or eventually at the delivery point. Such problem is called a multi-stage inspection problem.

* Corresponding author.

E-mail addresses: azadeh@ut.ac.ir (A. Azadeh), mssangari@ut.ac.ir (M.S. Sangari), ashamekhi@ut.ac.ir (A.S. Amiri).

Traditionally, the multi-stage inspection problem consists of a decision schedule in which some manufacturing stages receive full inspection and the rest none [4]. However, the problem is more complicated in a real serial multi-stage processing environment. In this case, three parameters are decided as the followings:

1. Whether inspection should be performed after each stage of the process or not.
2. The acceptance limits for each inspection station, i.e. tolerance of inspection.
3. The degree of inspection or size of sample to inspect.

Thus, the problem is to find the most efficient combination of three above decision parameters minimizing the total cost of inspection. The total inspection cost clearly depends on the number of inspection occurred in the production process and the number of defect products delivered to customers. In current research, a particle swarm optimization (PSO) algorithm is proposed to optimize such multi-stage inspection problem in which all three mentioned decision parameters is considered.

The paper is organized as follows: first, the previous works on inspection optimization problem is reviewed in Section 2. Then, the optimization problem in current work is described and formulated in Section 3. The solution algorithm based on PSO is introduced in Section 4. Section 5 illustrates application of the proposed algorithm in a numerical example and provides discussion on the results. The paper finally ends up in Section 6 with conclusions and some hints for future research.

2. Literature review

The process quality improvement using appropriate optimization methodologies has been a continual research endeavor [5]. It has been focused and investigated from different viewpoints in several previous works. Researchers have either applied exact solution approaches such as dynamic programming and integer programming, or approximate methods.

It has been demonstrated in early studies that for unconstrained systems and linear cost functions the optimal inspection policy at each of the inspection stations installed is 100% inspection Lindsay and Bishop [6] and White [7]. In more recent works in the field, Shiau [8] has assumed the limited number of inspection stations of each inspection station class, for solving the allocation problem in an advanced manufacturing system with multiple qualities characteristic. A unit cost model has been introduced considering the manufacturing capability, inspection capability, and tolerance specified concurrently for a multiple quality characteristic product. The situation of unbalanced tolerance design has been also investigated. Since determining the optimal inspection allocation plan seems to be impractical as the problem size becomes large, two decision criteria, i.e. sequence order of workstation and tolerance interval, have been employed separately to develop two different heuristic solution methods. The case has been further investigated by Shiau [9] to find a feasible inspection allocation plan where the specified tolerance of each quality characteristic varies from time to time according to the changing requirements of various customers.

Rau and Chu [10] have discussed inspection allocation problems for serial production systems with two types of workstations: attribute data and variable data, which better represents real practice. Three possibilities for the treatment of detected nonconforming units, namely, repair, rework and scrap have been considered. They have developed a profit model for optimally allocating inspection stations and introduced a heuristic solution method. In addition, Rau et al. [11] have developed a mathematical model considering layered fabrication to find the optimal solution for allocating inspections in re-entrant production systems. Workstations of variables data and inspections of quality characteristics measurement have been modeled and repair, rework, and scrap considered as three possibilities for the treatment of detected non-conforming units. Rau and Cho [12] have also addressed this problem and a genetic algorithm approach has been taken as the solution approach.

Wang [13] has focused on a production system subject to random deterioration where product inspections are performed only at the end of the production run. A mathematical formula representing the expected total cost per item has been derived as the objective function for such a system, where the in-control period follows a general probability distribution. Then, minimizing this objective function has been considered through the joint optimization of the production run length and product inspection policy. In another study, Wang and Yeh [14] have proposed an approximate production and inspection solution under the condition that the optimal inspection policy is equally-spaced. That is, an approximate production run length and number of inspections are obtained. The study has further investigated this approximate solution demonstrated that how to utilize it to obtain the real optimal solution more efficiently.

As a comprehensive inspection policy for serial multi-stage processes, considering all three previously mentioned inspection parameters concurrently will result in a complex joint optimization problem. To our knowledge, such an optimization problem has been only studied by Van Volsem et al. [15] and Azadeh and Sangari [16]. However, they both have considered the degree of inspection, i.e. size of sample, as a constant value not a decision variable to optimize.

Van Volsem et al. [15] have embedded a discrete event simulation to model the serial multi-stage process and introduced a genetic algorithm for numerical optimization purposes. They considered three inspection types in each work station, i.e. no inspection, sampling with a constant sample size, and full inspection. Azadeh and Sangari [16] have proposed another solution algorithm for the problem using simulated annealing (SA).

Table 1 provides a comprehensive overview of previous works in the field and main characteristics of each model.

In this paper, the problem of inspection optimization in serial multi-stage processes is expanded by including the size of sample as another decision parameter in the optimization model. Then, a solution algorithm using particle swarm optimi-

Table 1

Comprehensive overview of previous works in the field.

Author(s)	Main characteristics of the model
Lindsay and Bishop [6] and White [7]	<ul style="list-style-type: none"> • Optimal policy at each inspection station • Unconstrained systems • Linear cost functions
Klimberg et al. [21]	<ul style="list-style-type: none"> • A two-objective, zero-one programming model for inspection allocation problem
Villalobos et al. [22]	<ul style="list-style-type: none"> • Automated inspection strategies for production of printed circuit boards
Taneja and Viswanadham [23]	<ul style="list-style-type: none"> • Inspection location problem with manufacturing and scrapping and penalty cost • Genetic algorithm as solution approach
Bai and Yun [24]	<ul style="list-style-type: none"> • Optimal inspection level in a serial multi-stage production system • Location of limited number of automatic inspection • Exact solution algorithm for small size problems • A heuristic method based on backward dynamic programming as solution approach for large size problems
Barad and Braha [25]	<ul style="list-style-type: none"> • Optimal acceptance limits • Multi-stage process • Inspection is done at each production stage
Emmons and Rabinowitz [1]	<ul style="list-style-type: none"> • Assignment and scheduling of inspection tasks
Heredia-Langner et al. [4]	<ul style="list-style-type: none"> • Highly constrained multi-stage inspection problem • All stages must receive partial rectifying inspection • Real-valued genetic algorithm as solution approach
Kogan and Raz [26]	<ul style="list-style-type: none"> • N-stage production system with inspection activities • Problem of managing the intensity, sequence and timing of inspection • Minimizing sum of inspection costs and penalties caused by undetected defects
Shiau [8]	<ul style="list-style-type: none"> • Inspection allocation problem in an advanced manufacturing system • Limited number of inspection stations • Multiple quality characteristic • Unit cost model and unbalanced tolerance design • Heuristic solution methods
Shiau [9], Rau and Chu [10]	<ul style="list-style-type: none"> • Adding time dependency to tolerance design in Shiau [8] • Inspection allocation problem • Serial production system • A heuristic solution method
Rau et al. [11]	<ul style="list-style-type: none"> • Inspection allocation problem • Re-entrant production systems • Mathematical modeling considering layered fabrication
Wang [13]	<ul style="list-style-type: none"> • A production system subject to random deterioration • Inspections only at the end of the production run • Mathematical formulation of expected total cost per item • Joint optimization of production run length and product inspection policy
Freiesleben [2]	<ul style="list-style-type: none"> • Inspection allocation problem • Uniform defect rates • Genetic algorithm as solution approach
Van Volsem et al. [15]	<ul style="list-style-type: none"> • Serial multi-stage process • Joint optimization of inspection type and acceptance limits with a fixed sample size • Genetic algorithm as solution approach and Discrete event simulation
Rau and Cho [12]	<ul style="list-style-type: none"> • Genetic algorithm as solution approach for Rau et al. [11]
Wang and Yeh [14]	<ul style="list-style-type: none"> • An approximate production and inspection solution • Assuming that optimal inspection policy is equally-spaced
Azadeh and Sangari [16]	<ul style="list-style-type: none"> • Simulated annealing as solution approach for Van Volsem et al. [15]

zation (PSO), a metaheuristic method, is proposed. The efficiency and performance of the proposed algorithm is evaluated through comparing the results obtained from algorithms introduced by Van Volsem et al. [15] and also Azadeh and Sangari [16].

3. Problem statement

Consider a production process consisting of n consecutive process stages illustrated in Fig. 1. The products travel sequentially from stage 1 to n and the output of each process stage is the input for the next one. Thus, the last stage is dependent on the outputs of all former stages. The main benefit of inspection is that the downstream operations are not applied to already defective products, resulted in saving cost and preventing a congestion of the production flow.

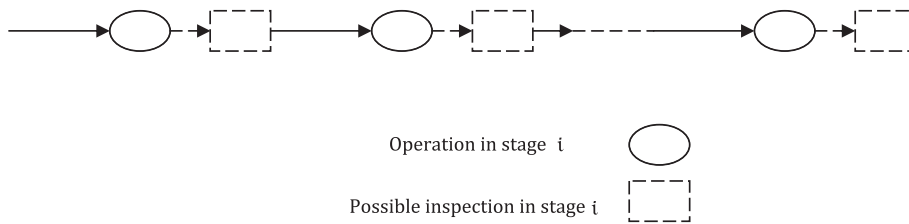


Fig. 1. The serial multi-stage process.

In such a process, the inspection system consists of m inspection stations such that $m \leq n$. At each stage, a decision should be made of the inspection extent. Two inspection types are assumed:

1. No inspection (N): the products go to the next stage immediately.
2. Inspection (I): a number of products (may be all of them) in the batch are inspected.

In the first inspection type, there is no more decision; but in the second type the decision should be made of inspection limits or products specification. For the sampling inspection, two other decision variables should be also identified, the sample size and acceptance number. Every product identified as defective through inspection should be reworked. It is further assumed that no inspection mistakes such as false reject or false accept errors occur.

To represent the mathematical model of the problem, it is assumed that there is a batch of K products and three types of inspection, namely no inspection (N), full inspection (F), and sampling (S). It is also assumed that X_i represents the decision about the extent of inspection or inspection type in stage i , i.e. $X_i \in \{N, F, \text{ and } S\}$. Where the decision is of sampling type, two more parameters are included: S_i as the sample size and a_i as the acceptance number for stage i , $i \in \{1, 2, \dots, n\}$. If the number of bad items in the sample of stage i (b_i) is greater than a_i , then the batch is rejected. In this case, a full inspection of the rejected batch is performed consecutively in the same stage.

In current research, the product dimension is considered as the quality characteristic for the serial n -stage process and tested at each inspection stage. The inspection limits for each stage i are as the followings:

LIL_i : lower inspection limit for stage i , and
 UIL_i : upper inspection limit for stage i .

Assuming the above notations, a product in stage i is considered as defective if the dimension value of this product lies outside the interval $[LIL_i, UIL_i]$. It is obvious that the lower product specification (LPS) and upper product specification (UPS) are equivalent to LIL_n and UIL_n , respectively.

Therefore, the total inspection cost (TIC) for every inspection policy consists of three cost components as the followings:

1. Inspection cost in stage i (IC_i): the cost of doing inspection, test or analysis in stage i .
2. Rework cost in stage i (RC_i): the cost of reworking products which are identified as defective through inspection or replace it with a non-defective product in stage i , and
3. Penalty cost (PC): the cost of delivering bad products to the customer. If the dimension value of the final product which is delivered to the customer lies outside of $[LPS, UPS]$, then a penalty cost incurs.

In addition, unit inspection cost in stage i , unit rework cost in stage i , and unit penalty cost are represented by ic_i , rc_i , and pc_i , respectively.

It is obvious that as the product flows to next stages the rework cost increases, i.e. for $i > j$ we have $rc_i < rc_j$. Since it is not economical that inspection be more expensive than rework, it is assumed that $ic_i < rc_i$. Thus, the total inspection cost can be determined with Eq. (1):

$$TIC = IC + RC + PC. \quad (1)$$

In other words, the total cost for an inspection policy is given by summation of total test and analysis cost, total rework cost, and total penalty cost where:

$$IC = \sum_{i=1}^n IC_i, \quad (2)$$

$$RC = \sum_{i=1}^n RC_i. \quad (3)$$

According to Eq. (2), total cost of performing inspection is calculated by adding the cost of performing inspection in all n stages of the process. Such a statement can be also given for Eq. (3) where total rework cost of an inspection policy is calculated. Moreover, IC_i and RC_i , the inspection and rework cost in stage i , are calculated by the following equations:

$$IC_i = \{0, \text{ if } X_i = N; ic_i^* K, \text{ if } X_i = F \text{ or } (X_i = S \text{ and } b_i > a_i); ic_i^* s_i, \text{ if } X_i = S \text{ and } b_i \leq a_i\}, \quad (4)$$

$$RC_i = \{0, \text{ if } X_i = N; rc_i^* d_i, \text{ if } X_i = F \text{ or } (X_i = S \text{ and } b_i > a_i); 0, \text{ if } X_i = S \text{ and } b_i \leq a_i\}, \quad (5)$$

where d_i is number of defective items in stage i . Eqs. (4) and (5) determine the related costs in three following conditions:

1. No inspection in stage i .
2. Full inspection in stage i or sampling inspection, where the number of bad items is more than the acceptance number.
3. Sampling inspection in stage i , where the number of bad items is less than the acceptance number.

The total penalty cost is also formulated as in Eq. (6):

$$PC = pc_i^* d_n, \quad (6)$$

where d_n is number of defective final products in the batch.

To have a better understanding, consider no inspection (N) as a feasible inspection policy to the problem. It is clear that adopting this policy results in minimum inspection and rework cost, but also yields maximum penalty cost. On the other side, full inspection (F) in all stages, leads to minimum penalty cost and causes that total inspection and rework cost to be maximized. These two conflicting inspection policy results in minimum value for each of the cost constituents separately. However, we should find an inspection policy which minimizes TIC as an aggregate cost function. Such a policy is considered as the optimal solution and also called an efficient inspection policy.

As described before, every inspection policy comprises a set of decision parameters: the number and location of inspection stations or inspection types, i.e. $X_i \in \{N, F, \text{ and } S\}$; the rigor of inspection or inspection limits, i.e. LIL_i and UIL_i for each $i \in \{1, 2, \dots, n\}$; and inspection extent, i.e. s_i and a_i . We consider a_i as a fixed parameter equal to 1, herein. Thus, the optimal inspection policy found by our proposed algorithm will offer optimal value for the following decision parameters:

1. $X_i \in \{N, F, \text{ and } S\}$.
2. LIL_i and UIL_i for each $i \in \{1, 2, \dots, n\}$, and
3. s_i .

A solution algorithm is proposed to find the optimal inspection policy including above inspection parameters to obtain the minimum total inspection cost.

4. Solution algorithm

It is clear that a serial process with n stages offers 2^n possible inspection combinations (whether to inspect or not in each stage). The complete enumeration of all combinations becomes more prohibitive as the number of stages increases. Therefore, application of metaheuristic methods will be more efficient to develop a solution algorithm as they need limited computational effort while yielding a nearly optimal solution. Herein, the solution algorithm is proposed based on PSO.

4.1. Particle swarm optimization

Particle swarm optimization (PSO) is an adaptive population based and derivative-free method, which is basically designed for continuous space optimization developed in 1995 [17]. It is inspired of social behavior of bird flocking and fish schooling. PSO is recently applied in many fields because of its simple structure with few numbers of parameters, which simplifies coding of the algorithm.

Suppose a swarm of birds searching for food in a space where there is only one piece of food available. Each particle's location in the multi-dimensional problem space is a feasible solution to the problem, which is evaluated with a fitness function. A particle in the swarm flies through the space near to the best own flying experience and swarm's flying experience. In other word, the strategy of the particle to find the food is changing the velocity to fly near the best place that has already experienced. PSO actually uses both aspects of cooperation and competition among the individuals in the population, which means it combines local and global search to reach the global optima. The distance of the particles to the food is measured by the pre-determined fitness function in all iterations. The particles in a local neighborhood share their information of their "best" positions, and then use the information to adopt their own velocities, and thus update their positions. In fact, each particle in this swarm has two kinds of intelligence, namely self-intelligence and social-intelligence. It is expected that the particles move towards better solutions in the feasible space.

PSO is a modern evolutionary algorithm comparable with genetic algorithm (GA). It is similar to GA in some aspects, such as starting with a randomly generated population (solutions), having a fitness function to evaluate the solutions, and using random techniques to update the population in all iterations. However in PSO, unlike GA, updating the particles depends on their memory and so does not have special operators [18]. It is also important to notice: "It has a more global searching abil-

ity at the beginning of the run and a local search near the end of the run. Therefore, while solving problems with more local optima, there are more possibilities for PSO to explore local optima at the end of the run" [19].

4.2. Basic PSO algorithm

In the literature of PSO, population is called swarm and each individual solution is called particle. The algorithm starts with a random population and then searches the feasible space for optima by updating the particles.

As mentioned before, self-intelligence is one of the critical parameters of PSO, which means each particle keeps its best position. The algorithm uses this as a factor for speed adjusting which is similarly relative to the global best position. In other words, while scanning the surface by its experiences, each particle sees the global best position and then move on the next iteration. So if the particle is far from the global best position, a higher change in its speed and direction is expected. The factors used in the algorithm are as the followings:

- x_k^i : position of i th particle in k th iteration.
- v_k^i : velocity of i th particle in k th iteration.
- p_k^i : the best ever position of i th particle from start to in k th iteration.
- p_k^g : global best position of swarm from start to in k th iteration.

The position of particle is updated in all iterations with following formula:

$$x_{k+1}^i = x_k^i + v_{k+1}^i, \quad (7)$$

while the velocity is updated with:

$$v_{k+1}^i = \omega_k \cdot v_k^i + c_1 r_1 (p_k^i - x_k^i) + c_2 r_2 (p_k^g - x_k^i). \quad (8)$$

In the above formula, r_1 and r_2 are random numbers distributed uniformly between 0 and 1. These two parameters moderate the effect of p_k^i and p_k^g on velocity. c_1 and c_2 are constant values representing the degree of importance of p_k^i and p_k^g . For example, if c_2 is set greater than c_1 , the global best position has greater degree of importance. ω_k is inertia of the particle which is to keep the particle moving and reacts on the capability of overall balance and part searching [20]. ω_k can be set as a constant value or a variable changing in all iterations.

The general algorithm of basic PSO is shown in Fig. 2. As it can be seen from the flowchart, implementing the algorithm to the optimization problems is so easy with respect to other similar techniques. In current research, it is used to construct a solution algorithm for the inspection optimization problem.

4.3. Proposed solution algorithm

As previously described, there are both continuous and discrete variables in the inspection optimization problem. Integer variables are those which determine what kind of inspection should be done in each stage ($X_i \in \{N, F, \text{and } S\}$) and also the sample size (S_i). In addition, tolerance of inspection is the continuous variable of the problem (LIL_i and UIL_i for each $i \in \{1, 2, \dots, n\}$). It determines the acceptable range for the dimension of parts to be inspected.

There are some challenges for solving such problem. Controlling these variables, especially the mentioned continuous variables, is much difficult because their "feasible" interval is absolutely tiny. The term "feasible" herein means that the range for inspection tolerance has a reasonable limit. For example, it is not reasonable to specify the tolerance ± 5 for a part with the length of 10. Thus, it is necessary to tune the PSO parameters in a way that the solutions in all iterations not escape from a particular number. If the parameters are not tuned finely, the solutions may escape from this number and disturb the problem solving procedure. Moreover, this number is not pre-determined; it is inferred from the nature of the problem. Therefore, the decision maker can tune a fine number for the PSO parameters after getting insight to the problem. For the current problem, it is inferred that the tolerance should not be so far from $2 + \varepsilon$.

Before describing how the initial solution is generated, let us discuss the solution method a bit more. As mentioned before, three different situations may happen in each process stage which is shown by X_i : F (full inspection), S (sampling), and N (no inspection). A straight way to solve the problem is to map these integer variables to a continuous range of 0 to $15 - \varepsilon$ where ε is a small positive value. Then, one of the following rules will be applied for determining the value of X_i :

- If $X_i \geq 0$ and $X_i < 5$, then no inspection is done.
- If $X_i \geq 5$ and $X_i < 10$, then sampling inspection is done.
- If $X_i \geq 10$, then full inspection is done.

As stated before, the sample size has been treated as a constant value not a decision parameter in previous works. However, it is another integer variable herein and should be optimized with the solution algorithm. From the operator's point of view, if sampling inspection is adopted, i.e. $X_i = S$, then he should select s_i parts randomly and check them. We set the initial

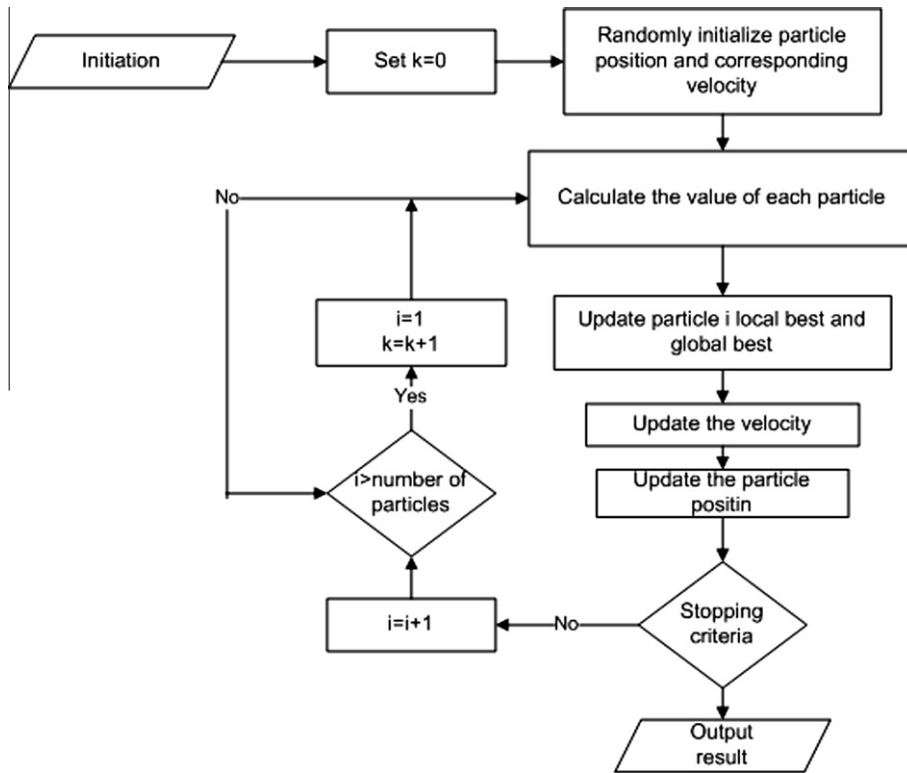


Fig. 2. Flow chart for basic PSO.

value of s_i equal to 50 in the solution procedure (equal to its constant value in previous works), but it will change during the algorithm iterations.

Furthermore, for the standard continuous variables, i.e. the tolerance of inspection which has been described earlier, the “Floor” operator is used to map them to the integer.

4.3.1. Generating an initial solution

To generate an initial solution, a uniformly distributed function between 0 and 15 is applied for X_i . After the type of inspection for the particles is determined, the sample size is set at 50 for all of them. As the sample size is variable, it will change in next iterations.

To determine the tolerance of inspection, a function like $2 + \varepsilon$ is defined in which ε is of uniform function between -0.5 and 0.5 .

After determining all the necessary parameters, then all solutions will be evaluated with a fitness function. The current solution for all of the particles is set to the best position of the particle and the best solution is set to the best global.

4.3.2. Parameter setting and updating the particles

In this phase, the most important thing is to set the parameters appropriately. Although ω_k is a constant value in basic PSO, but we can update it through the next iterations. For this purpose, we have:

$$\omega_k = \omega_{max} - \left(\frac{\omega_{max} - \omega_{min}}{k_{max}} \right) \times k \quad (9)$$

in which ω_{max} and ω_{min} are the maximum allowable inertia and k_{max} is the maximum number of iterations. The inertia will be updated through the iterations such that in early iterations has the greater value and at the end has the minimum value. It is owing to the fact that in early iterations it is needed to search more and in last iterations it is needed to diminish the steps.

The parameters ω_{max} , c_1 , and c_2 for LIL_i and UIL_i , are set at a very small value, for example 0.05, because the large number for coefficients may cause diversity in newly generated solutions. Although another straight method is to widen the solution space.

5. Discussion and analysis

To ensure that the proposed algorithm works properly and yields efficient solutions, a numerical example is investigated and the obtained results are discussed in two phases.

5.1. Numerical example characteristics

The required data the problem characteristics are adopted from the numerical example discussed in Van Volsem et al. [15]. Table 2 shows the characteristics of this problem.

This serial multi-stage process consists of six stages. Production in stages 1, 2, 4 and 5 follows normal distributions and in stages 3 and 6 keeps uniform distributions. Parameter 1 is the mean value for normal distributions and is the lower bound for uniform distributions. The second parameter also states the standard deviation for normal distributions and the upper bound for uniform ones. The expected value of the product dimension at each stage is presented in the fifth column. The last two columns of the table show the inspection, test or analysis cost and rework cost at each stage, respectively. It is obvious that when a product flows in the system from stage i to j , the cost of rework then increases and touches the maximum amount at final stage of the process.

The penalty cost of delivering a defective product to the customer is 3000 in this case and the acceptance number in a sample (a_i) is set at 1. A batch with 1000 products is also assumed. The final product will be accepted if its dimension lies in the interval [58,62].

5.2. Sample size as a fixed value

First, the problem is solved with a fixed sample size. Similar to the studies by Van Volsem et al. [15] and Azadeh and Sangari [16], the sample size is set at 50. The results of 10 runs of the algorithm are given in Table 3.

The final result after several runs of the algorithm is NXFNXF. According to this string, no inspection is planned for stages 1 and 4 and full inspection is planned for stages 3 and 6. Moreover, there is not a unique decision for stages 2 and 5. It is denoted by X where $X \in \{N, S\}$ based on the results given in Table 3.

The general answer from the proposed algorithm is rather different from the results obtained from two earlier works in which the string of NNFXFX is the optimal answer. However, the algorithm developed herein based on PSO results in a better answer as it yields a smaller objective value. The average total inspection cost (TIC) as the objective value obtained from the proposed solution algorithm after several runs is 123,075, while this measure for the genetic algorithm proposed by Van Volsem et al. [15] and simulated annealing approach proposed by Azadeh and Sangari [16] is 125975.8 and 125747.4, respectively. Therefore, PSO gives better results.

Table 2
Characteristics of the numerical example.

Stage no.	Distribution	Parameter 1	Parameter 2	Expected value	Inspection or test cost	Rework cost
1	Normal	10	0.3	10	1	50
2	Normal	10	0.5	20	1	100
3	Uniform	8.5	11.5	30	2	200
4	Normal	10	0.1	40	1	400
5	Normal	10	0.5	50	1	800
6	Uniform	9	11	60	2	1600

Table 3
Obtained results for the numerical example using proposed solution algorithm where sample size is a fixed value ($s_i = 50$).

Number of run	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5	Stage 6	Objective value
1	N	N	F ^{1.7692}	N	N	F ^{2.0228}	120,500
2	N	N	F ^{1.8852}	N	N	F ^{1.9877}	123,800
3	N	N	F ^{1.9173}	N	S ^{1.9656}	F ^{1.9811}	120,750
4	N	S ^{1.247}	F ^{1.8195}	N	N	F ^{1.8692}	123,550
5	N	N	F ^{1.9545}	N	S ^{1.7386}	F ^{1.9880}	121,850
6	N	N	F ^{1.8303}	N	N	F ^{2.0477}	126,600
7	N	N	F ^{1.8076}	N	N	F ^{2.0181}	125,400
8	N	S ^{1.4494}	F ^{1.7916}	N	N	F ^{1.9794}	123,250
9	N	S ^{1.4994}	F ^{1.7687}	N	N	F ^{1.9901}	122,850
10	N	N	F ^{1.9155}	N	N	F ^{2.0258}	122,200

N: no inspection; F^x: full inspection with tolerance of x ; and S^x: sampling inspection with tolerance of x .

Table 4

Obtained results for the numerical example using proposed solution algorithm where sample size is a decision parameter.

Number of run	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5	Stage 6	Objective value
1	<i>N</i>	<i>N</i>	$S_{92}^{1.2783}$	<i>N</i>	<i>N</i>	$F^{1.9873}$	120,684
2	<i>N</i>	<i>N</i>	$S_{98}^{1.3300}$	<i>N</i>	<i>N</i>	$F^{2.0035}$	118,096
3	<i>N</i>	<i>N</i>	$S_{97}^{1.2234}$	<i>N</i>	<i>N</i>	$F^{1.9937}$	123,094
4	<i>N</i>	<i>N</i>	$S_{84}^{1.1703}$	<i>N</i>	<i>N</i>	$F^{2.0315}$	118,668
5	<i>N</i>	<i>N</i>	$S_{97}^{1.5563}$	<i>N</i>	<i>N</i>	$F^{2.0532}$	118,094
6	<i>N</i>	<i>N</i>	$S_{92}^{1.2435}$	<i>N</i>	<i>N</i>	$F^{2.0103}$	121,394
7	<i>N</i>	$S_{81}^{1.2547}$	<i>N</i>	$S_{63}^{1.4504}$	<i>N</i>	$F^{2.0241}$	119,844
8	<i>N</i>	$S_{78}^{1.1369}$	<i>N</i>	$S_{64}^{1.304}$	<i>N</i>	$F^{2.0198}$	122,442
9	<i>N</i>	$S_{96}^{1.3110}$	<i>N</i>	$S_{61}^{1.1158}$	<i>N</i>	$F^{2.9835}$	120,457
10	<i>N</i>	$S_{80}^{1.2547}$	<i>N</i>	$S_{73}^{1.2041}$	<i>N</i>	$F^{2.0014}$	119,153

N: no inspection; F^x : full inspection with tolerance of x ; and S^x : sampling inspection with tolerance of x and sample size of s .

It should be noted that the proposed algorithm has a difference with the two earlier models. They have calculated the ratio of defected items based on the distribution function of production, but in the proposed PSO method the number of defected products is calculated based on defected number of parts that produced via simulation.

5.3. Sample size as a decision variable

At the second phase, the primary model is extended with a new decision parameter. The numerical problem is then solved with a variable sample size. The results of 10 runs of the algorithm are shown in Table 4.

As given in Table 4, the optimal solution is changed to $NXXXNF$ where $X \in \{N, s\}$. According to the results, no inspection is the optimal decision for stages 1 and 5. In addition, full inspection is planned for the sixth stage.

Furthermore, it is realized that the inspection policy for stage 2–4 denoted by XXX will be of two sets: NSN (solutions 1–6 in Table 4) and SNS (solutions 7–10). In other words, if no inspection is planned for stage 2 the inspection in stage 3 will be then of sampling type and no inspection is planned for stage 4. Moreover, where inspection in stage 2 is of sampling type no inspection is then happened to stage 3. In this case, stage 4 also requires sampling inspection. From a theoretical point of view, when the weight of global optimum in the PSO relations are greater or equal to the local optimum, the first solution set is then obtained, but when the coefficients for local optimum is significantly greater than the coefficients of the global optimum, the second set is obtained. However, the objective values for both sets of solutions are quite close to each other.

Considering the sample size as a decision parameter to be optimized reduces the average of objective function value (TIC) to 120192.6 and leads to a better performance. The full inspection is not needed in third stage in this scenario. Moreover, the sample size of 50 is not enough. However, the last stage needs full inspection in all of the solutions.

Note that it is reasonable that the inspection in the last stage is done by the tolerance of 2. The tolerance in the last stage is clearly close to 2.

6. Conclusions

As inspection systems requires valuable organizational resources and incurs cost to the organizations, it is important to find an efficient and economical inspection policy to implement. This paper has studied the optimal inspection policy for inspection in a serial multi-stage process. Three decision components have been assumed for such policy: determining the stages in which inspection should occur and type of inspection (full or sampling), tolerance of inspection (acceptance limits), and the sample size where sampling inspection is decided. The product dimension is considered as the quality characteristic to be tested at the inspection stages as the quality characteristic for the considered serial n -stage process.

It is clear that a process including n serial stages offers 2^n possible inspection combinations. As the number of stages increases, complete enumeration of all combinations becomes more prohibitive. Therefore, application of metaheuristic methods is more efficient to develop a solution algorithm, as they need limited computational effort while yielding a nearly optimal solution. In this paper, a solution algorithm based on PSO has been developed to achieve the optimal inspection policy.

The algorithm has been investigated in two phases using a numerical example adopted from earlier works. In the first phase, the size of sample to inspect has been set at a fixed value similar to previous works. Secondly, the model has been extended and the sample size has been considered as a decision parameter which should be optimized. The obtained solutions in both cases have validated the practicality and efficiency of the algorithm. Moreover, it has been also proved that the proposed solution approach using PSO gives better quality results in comparison with previous algorithms considering total inspection cost as the performance measure.

The proposed approach is especially helpful for production and quality managers dealing with the problem of allocating inspection facilities in sequential processes, such as the assembly lines, and provides the practitioners with an economical

inspection policy. The results can be used as a basis for further exploration and discussion by the management to make the final decision regarding the inspection system.

Acknowledgement

The authors are grateful for the valuable comments and suggestion from the respected reviewers. Their valuable comments and suggestions have enhanced the strength and significance of our paper.

This study was supported by a Grant from University of Tehran (Grant No. 8106013/1/05). The authors acknowledge the supports provided by the University College of Engineering, University of Tehran, Iran.

References

- [1] H.E. Emmons, G. Rabinowitz, Inspection allocation for multistage deteriorating production systems, *IIE Trans.* 34 (2002) 1031–1041.
- [2] J. Freiesleben, Costs and benefits of inspection systems and optimal inspection allocation for uniform defect propensity, *Int. J. Quality Reliab. Manage.* 23 (2006) 547–563.
- [3] S. Van Volssem, A method for determining cost-efficient inspection strategies in multistage production systems, *4OR: Quart. J. Oper. Res.* 5 (2007) 343–346.
- [4] A. Heredia-Langner, D.C. Montgomery, W.M. Carlyle, Solving a multistage partial inspection problem using genetic algorithms, *Int. J. Prod. Res.* 40 (2002) 1923–1940.
- [5] I. Mukherjee, P.K. Ray, Quality improvement of multistage and multi-response grinding processes: an insight into two different methodologies for parameter optimization, *Int. J. Prod. Quality Manage.* 4 (2009) 613–643.
- [6] G.F. Lindsay, A.B. Bishop, Allocation of screening inspection effort – a dynamic programming approach, *Manage. Sci.* 10 (1964) 342–352.
- [7] L.S. White, Shortest route models for the allocation of inspection effort on a production line, *Manage. Sci.* 15 (1969) 249–259.
- [8] Y.-R. Shiau, Inspection allocation planning for a multiple quality characteristic advanced manufacturing system, *Int. J. Adv. Manufact. Technol.* 21 (2003) 494–500.
- [9] Y.-R. Shiau, Quick decision-making support for inspection allocation planning with rapidly changing customer requirements, *Int. J. Adv. Manufact. Technol.* 22 (2003) 633–640.
- [10] H. Rau, Y.-H. Chu, Inspection allocation planning with two types of workstation: WVD and WAD, *Int. J. Adv. Manufact. Technol.* 25 (2005) 947–953.
- [11] H. Rau, Y.-H. Chu, K.-H. Cho, Layer modeling for the inspection allocation problem in re-entrant production systems, *Int. J. Prod. Res.* 43 (2005) 3633–3655.
- [12] H. Rau, K.-H. Cho, Genetic algorithm modeling for the inspection allocation in reentrant production systems, *Expert Syst. Appl.* 36 (2009) 11287–11295.
- [13] C.-H. Wang, Integrated production and product inspection policy for a deteriorating production system, *Int. J. Prod. Econ.* 95 (2005) 123–134.
- [14] C.-H. Wang, R.H. Yeh, Utilizing an approximative solution to obtain the real optimal solution for a production and inspection model, *Appl. Math. Model.* 33 (2009) 3769–3775.
- [15] S. Van Volssem, W. Dullaert, H. Van Laneghem, An evolutionary algorithm and discrete event simulation for optimizing strategies for multi-stage processes, *Eur. J. Oper. Res.* 179 (2007) 621–633.
- [16] A. Azadeh, M.S. Sangari, A metaheuristic method for optimizing inspection strategies in serial multi-stage processes, *Int. J. Prod. Quality Manage.* 6 (2010) 289–303.
- [17] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: *Proceedings of IEEE International Joint Conference on Neural Networks*, Piscataway, NJ, vol. 4, 1995, pp. 1942–1948.
- [18] A.N. Haq, K. Karthikeyan, K. Sivakumar, R. Saravanan, Particle swarm optimization (PSO) algorithm for optimal machining allocation of clutch assembly, *Int. J. Adv. Manufact. Technol.* 27 (2006) 865–869.
- [19] S. Önüt, U.R. Tuzkaya, B. Doğan, A particle swarm optimization algorithm for the multiple-level warehouse layout design problem, *Comput. Ind. Eng.* 54 (2008) 783–799.
- [20] L. Juan, J. Hui, Fuzzy inspection of fabric defects based on particle swarm optimization (PSO), *Lect. Notes Comput. Sci.* (2008) 700–706.
- [21] R. Klimberg, C. ReVelle, J. Cohon, A multiobjective approach to evaluating and planning the allocation of inspection resources, *Eur. J. Oper. Res.* 52 (1991) 55–64.
- [22] J.R. Villalobos, J.W. Foster, R.L. Disney, Flexible inspection systems for serial multi-stage production systems, *IIE Trans.* 25 (3) (1993) 16–26.
- [23] M. Taneja, N. Viswanadham, Inspection allocation in manufacturing systems: a genetic algorithm approach, in: *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, 1994, pp. 3537–3542.
- [24] D.S. Bai, H.J. Yun, Optimal allocation of inspection effort in a multi-stage production system, *Comput. Ind. Eng.* 30 (1996) 387–396.
- [25] M. Barad, D. Braha, Control limits for multi-stage manufacturing processes with binomial yield (single and multiple production runs), *J. Oper. Res. Soc.* 47 (1996) 98–112.
- [26] K. Kogan, T. Raz, Optimal allocation of inspection effort over a finite planning horizon, *IIE Trans.* 34 (2002) 515–527.