J Intell Manuf (2018) 29:405–422
https://doi.org/10.1007/s10845-015-1116-7

A Genetic Algorithm applied to pick sequencing for billing

Anderson Rogério Faia Pinto[1] · Antonio Fernando Crepaldi[2] · Marcelo Seido Nagano[1]

Received: 28 October 2014 / Accepted: 14 June 2015 / Published online: 26 June 2015

Abstract This article addresses the use of Holland's Genetic Algorithms (GAs) (Holland in Adaptation in natural and artificial systems, University of Michigan Press, Ann Arbor, MI, 1975) in solving an optimization problem not exploited yet by literature, which we have named Optimal Billing Sequencing (OBS). The objective of the GA proposed is to automate pick sequencing, which addresses the process of allocating the stock available for sale to the purchase orders in a portfolio, so that the maximization of the billing is the optimal result for the OBS. A modelling and computational simulation methodology has been employed. Such methodology is designed to enable the GA to meet the boundary conditions established by predefined decision restrictions and parameters. We have reached the conclusion, by means of experimental tests, that the GA developed satisfactorily solves the problem studied. In addition to a low computational overhead, the GA reduces operating costs and speeds picking decision-making processes and billing processes.

Keywords Genetic Algorithms · Picking process · Billing sequencing

✉ Marcelo Seido Nagano
drnagano@usp.br

Anderson Rogério Faia Pinto
anderson@life.com.br

Antonio Fernando Crepaldi
crepaldi@feb.unesp.br

1 Production Engineering Department, School of Engineering of São Carlos, University of São Paulo, Av. Trabalhador São Carlense, 400, São Carlos, SP 13566-590, Brazil

2 Production Engineering Department, Faculty of Engineering of Bauru, São Paulo State University, Av. Luiz Edmundo Carrijo Coube, 14-01, Bauru, SP 17033-360, Brazil

Introduction

The dynamics of changes in corporate environments and the search for adaptations that provide immediate answers for the competitive market are grounds on which to base the importance of actions that identify the minimum of options needed to restrict uncertainty. Hence, the practice of optimization is an indispensable element of competitiveness and, as stressed by Elsayed et al. (2014), it is essential in solving many problems. The conviction that the resources, time and money are limited have raised the importance of, and the demand for, more robust optimization techniques, even in the presence of analytically untreatable problems (Yang and Koziel 2010). Thus, the current management practice constantly looks for new technologies that are capable of rapidly and consistently presenting alternatives, which result in better corporate management. In the last decades, it has become common among researchers such as Bäck and Schwefel (1993), Blickle (1996), Mitchell and Taylor (1999), Runarsson and Jonsson (1999), Yang (2005), Nobakhti (2010), Zhang et al. (2012), for instance, to imitate evolutive genetic mechanisms to develop Evolutionary Algorithms (EAs) that assist managers in solving complex optimization problems.

Recently, inspired in the adaptation principles of the modern natural and biological evolution (please refer to Bowler 2001; Ghiselin 2009; Phillips and Su 2009 and Ayala 2010), the EAs are applied to a wide range of *NP-hard* problems of diverse areas in the corporate environment. Nowadays EAs represent the most recent innovation line that had a deep effect on science when applied to intractable problems (Nobakhti 2010; Zhang et al. 2012). Among the different EAs, Genetic Algorithms (GAs) is the most popular and widely used (Goldberg 1989). These distinguish by simplicity of operation, robustness and by the ease and flexibility of implementation, including by hybridization with conventional methods (McCall 2005; Yang 2005). Indicated to NP-hard they don't require many mathematical requirement, obtaining viable solutions with a cost of programming and computer time satisfactory when compared to other methods, specially, matching and sequencing problems, and its efficacy is attested by the big number of published papers (Hallam et al. 2010; Whitley and Sutton 2012; Elsayed et al. 2014).

Therefore, the purpose of this article is to develop a computational method, with a structure based on Holland's GA (Holland 1975), which presents an optimal solution for the OBS problem. Usually, the OBS consists in optimizing the ability to satisfy purchase orders received according to the finished products (*FP*) available and maximizing revenues. Thus, GA will work in form of attribution of each products and quantities will be invoiced to each client is the operation is herein denominated "picking process". In actual cases, OBS problems are mostly triggered by restrictions or flaws in the conception of managerial systems, attributed, in its majority, to disorders and uncertainties caused over time because of the dynamics of changes in, and the complexity of, the systems used by organizations. An element that aggravates the OBS is the worldwide tendency to reduce stock, since the purchase order portfolio (*POP*) is constantly renewed because of increases in the frequency of orders, in addition to possible changes or cancellations. Consequently, if at a given time the *FP* are not sufficient to bear the *POP*, the OBS may become a complex task to managers. That is because the attention centered in the restriction setting and internal guideline and also exigencies demanded by the clients makes that the optimization of the OBS to be subject to several assessments among the possible billing alternatives that exist.

In this paper, the implicit presumption to OBS are based in the paper of Rim and Park (2008), I which it's presupposed a typical industrial scenario, although, based in real standards. In fact, the literature is limited to models that presuppose restrictions of stock and picking planning. In the bibliographic investigation made only by Rim and Park (2008) the stock shortage to the attending problem of picking orders in distributions centers (DC) s treated. The author's aim is to attribute the stock to the orders in a way to maximize the order fill rate (OFR). In this case, the OFR is pondered and used to reflect the importance of service of each client/order. The purchase order is done if all items are available in stock and the not carried orders are transferred to the next day according to importance and priority in a way to avoid excessive delays. The linear programming (LP) totally binary is used as a method of solution and the performance of the model is compared to the first come, first served (FCFS) rule, in which the results overcome the simple rules already existing.

Although we did not find any authors using GA to solve OBS problems during the bibliographic research, GA is used to solve various problems addressed in GA-specific literature. Evidently, the existence of different organizations that integrate greatly varied markets, each of which with their own peculiarities and specific decision-making criteria, make it difficult to generalize the OBS problem. Hence, the conception of GA necessarily requires delimiting the OBS in a study that establishes certain particulars and decision-making criteria. In general terms, the boundary conditions and procedure rules are: (i) billing preference occurs in increasing order according to the date the purchase orders are effectively processed and picking may not take place either if there are not any *FP*s available or there is a only a partial number of *FP*s available, but the client does not accept such partial number; (ii) random binary representation generates attribution structures and the fitness function penalizes undue bits by their billing amounts; (iii) elitist selection transfers the best individual to the new population that linked to the crossover action and to mutation enables the GA to repeat evolution cycles to find, if not the optimal, the best solution until the stopping criterion is satisfied.

In methodological terms, this article is classified as an applied research in which a quantitative modelling approach and computational simulation are employed (Bertrand and Fransoo 2002; Haegeman et al. 2013). The programming will be developed in Microsoft Office Excel 2010's Visual Basic language for applications (VBA) and the data entry and results analyses will be carried by the Excel's spreadsheets themselves. The GA proposed will be implemented and subjected to computational experimentations in a computer powered by a 2.3 GHz Core i5 processor with 8 GB of RAM. This article has been structured in sections, summarized as follows: "Contextualization and rationale" section brings all the contextualization of the problem and the justification of research; "Optimal Billing Sequencing (OBS)" section describes in detail the problem studied; "Hybrid Genetic Algorithm" section contains a brief explanation of GAs and expresses the GA proposed; "Experimentation and discussion of the results" section addresses the experimentations and analysis of the results obtained; and "Final considerations" section closes the article with the final comments, main contributions of the research, and suggestions for future studies on the matter.

Contextualization and rationale

Into the business range, incertitude associated to management of demand is a critical parameter. According to Sereshti and Bijari (2013), foreseeing the demand in an exact way is impossible. The fact is that most companies operate in a complex and unstable environment, which makes accurate forecasting difficult (Baud-Lavigne et al. 2014). Furthermore, most of the quantitative tools that exist today is analytical instead of predictable and they are incapable of handling with the future (Haegeman et al. 2013) or yet introduce acceptable solutions in a short period of time. Expressing all the variability of the setting in precise equations in an unmistakable way is still impracticable. Options restrict in anticipating, when you know a priori, the state realized or at least react to unlikely events. According to Slotnick (2011), in a technical way, the selection of purchases depends on the coordination and capacity with the demand and the trade-off between the incomes and processing costs. Some more recent researches have used demanding taxes as a level function and time of stock cycle, obtaining increase in sales and maximization of medium profit by unit of time (Pando et al. 2012). Evidentially, one order is totally served when the quantity of all the required items are available in stock (Simchi-levi et al. 2003). However, if the production system's optimization depends on the market forecast, it's not always possible to satisfy all the potential demands to maximize the reception and profit (Shen 2006; Baud-Lavigne et al. 2014). The quantity of some products in stock, can, sometimes, be smaller than the totals demanded by the orders (Rim and Park 2008).

Conventional solutions to minimize these problems are in the extra hours, resources deviation, outsourcing or negotiation of deadline and prices (Slotnick 2011). However, deals are not always possible and some resources may not be available in moments of urgency. So, if the production mix should be the one who maximizes the profit (vide Chen et al. 2013; Wang and Dargahi 2013; Xiang et al. 2014 and Zhuang and Chang 2015), then, in real practical situations the referred approach attends, only in parts, the set of demands imposed by internal managers and external clients. In the current market the client's satisfaction is linked to total availability of the products that best adjust to their needs or overcome the expectations in terms of quality, price and deadlines facing the possible levels of speed and flexibility in the service. (Hiremath et al. 2013; Risdiyono and Koomsap 2013). It's also seen that due to the growing demands of the clients and levels of global competition there is a big pressure of internal managers to minimize the costs and define competitive prices keeping the high level of quality of the products and post-selling services (Mousavi et al. 2014; Jeang 2015). It's undeniable that the most reliable option and that could result in a better level of client service would be keeping sufficiently high level stocks to all the items on demand (Rim and Park 2008). This, however can lead to failures in productivity and monetary lost due to the increase of stock prices and eventual obsolesces, besides the restrictions of physical space to storage (Tompkins and Smith 1998; Rim and Park 2008).

It is fact that the globalization, the introduction and growth of online sales and the Lean Manufacturing and Just-in-Time philosophies, and the significant reduction of lead times, have taken the company to engage in reducing the orders cycles by eliminating all wastes and operations without aggregate price (Gu et al. 2007; Richards 2011; Haq and Boddu 2014). At the same time, the innovations related to actual cost methodologies, from supply chain management (SCM) and the theory of constraints paradigm among other specific practices such as the total quality management, Downzising and Kaisen, have made the minimization of stocks a world axiom. The companies, then, started to aim to achieve a big volume of production and distribution with minimum levels of stock and short time of answer (Van Berg and Zijm 1999). The recent notions that the productive systems may be able to react quickly to unexpected changes were intensified. The adaption to immediate demands of the competitive market started to impel future directions in the preview of new technologies directed to flexibility and optimization of the process. In order to attend the demand for strategies and stronger optimization methods advanced decision and intelligence technologies for manufacturing and logistics are stated by Chien et al. (2012). In general, the scientific community has proposed numerous studies to adaptation to the incertitude of the market and the fast demand variations aiming to optimize and integrate different manufacturing problems, inventory and SCM so that in an agile and flexible way, it minimizes costs and attend efficiently the client's orders. (consult Bandyopadhyay and Bhattacharya 2014; Baud-Lavigne et al. 2014; Diabat et al. 2015; Ghiami et al. 2013; Hiremath et al. 2013; Haq and Boddu 2014; Kumar et al. 2014; Mousavi et al. 2013, 2014; Park and Kyung 2014; Sadeghi et al. 2014; Triki et al. 2014; Xiang et al. 2014; Yao and Huang 2014; İnkaya and Akansel 15).

Facing this scenario, there was an expressive change in the picking operations in the last 20 years (Richards 2011). This is why, previously, the pallets and whole boxes picking was normal, but nowadays the clients started to require products in smaller quantities and more frequently, so, the orders were size-reduced and generated in shorter periods of time (Richards 2011). This prepossession made that the merchandise reposition and, consequently, the collocation, processing and shipment of orders become frequent, which, adding to eventual changes or cancelling, took *POP* to a constant renovation cycle (Richards 2011). Rim and Park (2008) and Matthews and Visagie (2013) point out that even if the range of products requested and also the diversification in the client's profiles, based on the quantity of sales, markup or marketing strategies, made that *POP*s with orders non uniforms and rankings to prioritization of service.

This way, in cases where the orders require different items and the stock is controlled to a minimum level it is not uncommon the occurrence of stock shortage to serve them (Rim and Park 2008 and Sereshti and Bijari 2013). Soon, the ideal service of an order can prevent the compliance of many others and, however, develop situations where OBS solutions can be found. When facing a big number of combinations undergoing a series of evaluations, among the possible alternatives of existing billing, the search for a better solution to OBS can become a complex task with which the managers have to handle. This implicit complexity to OBS is, in general, associated to policies and contour conditions established by restrictions, variables and parameters pre-defined attached to the company who defines as decision criteria, which generally involve: (i) client demands; (ii) quantity-related restrictions; (iii) number of purchase orders; and (iv) the diverse items ordered, among others. A typical example was quoted by Rim and Park (2008) referring to e-commerce, in which the clients usually don't accept to receive partial purchases and that these decisions are strongly related to matters of increase in the shipping expenses.

The application of exacts techniques of optimization assume that it is not a feasible option, since that, the solutions can demand intense computational effort and time, besides the risk of ineptitude to handle with the parameters and number of variables and restrictions. To Whitley and Sutton (2012) the GA's use as a solution technique is a proper option on these cases. It's clear that amidst many existing optimization paradigms the application of other techniques and hallowed heuristic could also be evaluated. Dynamic Programming, e.g., is an efficient method when used in the resolution of combined problems and Tabu Search is also other important approach at combined resolution of problems. However, according to Yang and Koziel (2010), algorithms are as varied as the optimization itself, what would demand substantial effort and studies to identify which is the most appropriate and this is not the purpose of this paper.

The search for optimized solutions for a specific OBS problem based in the search potential and practical efficacy from the GA's is the focus given to this paper. The perspective for solving the OBS applying the GA to the picking process, which aims at meeting the needs of the *POP* in view of the boundary conditions established by predefined restrictions and parameters so that the maximization of the billing is the optimized result for the OBS. When considering that managers need to take quick decision and that they should result in a series of consisting actions, it is believed to be of great value the development of a computational tool able to provide optimized solutions to a real problem not yet approached by literature. This way, the proposed GA substitutes the Rim and Park methods (2008) eliminating the risk of stock shortage after invoice and the not planning of picking orders.

Optimal billing sequencing (OBS)

This section describes the OBS problem in detail and *Rg* is the number that differentiates the n products in *FP*. Every *Rg* has a number of items defined by x_i, where the subscript $i = (1, 2, \ldots, n)$ denotes the ith Rg. If X is set x_i, then $X = \{x_1, x_2, \ldots, x_n\}$ represents the maximum number of *Rgs* available for sale at a given time t.*POP* refers to the purchase order portfolio, constituted by a set of *POs*—purchase orders—where the subscript j refers to the jthPO; then, $POP = \{PO_1, PO_2, \ldots, PO_n\} \forall j = (1, 2, \ldots, n)$, where n is the number of *POs*. Usually, a *POP* is updated every day in view of the PO_j cycles and the demand for Rg_i of PO_j is given by q_{ij}, represented by $PO_j = \{q_1, q_2, \ldots, q_n\}$, $\forall i = (1, 2, \ldots, n)$. Although they may contain similarities, each PO_j requests q_i that are the attributes of a certain client *Ca*, in which, the *GC* set represents the group of C_m clients; therefore, $GC = \{C_1, C_2, \ldots, C_m\}$, $\forall \alpha = (1, 2, \ldots, m)$, where m is the number of clients. Thus, C may contain $PO_j + 1$ in the *POP* in t and the sum of q_i of PO_n generates the total demand for q_i of the *POP* called Q_i.

Accordingly, q_i must be attributed to PO_j in a non-overlapping (disjoint) manner by comparing Q_i with supply x_i, where q_i is defined as the relevance of the ith product for the jth purchase order, such that $PO_1 \cup PO_2, \ldots, PO_j \leq x_i$. Such operation is called "picking process" and specifically addresses the way of designating q_i of Rg_i to be billed to PO_j of *Ca*. Nevertheless, effectively satisfying the *POP* involves analyzing a set of restrictions and decision-making criteria that are strongly related. In the OBS studied, the satisfaction requisite is in increasing order according to the date the purchase orders are effectively processed d, which requires a billing sequencing for PO_j, and the restrictions reflect the condition of x_i in satisfying Q_i. Therefore, if $x_i < Q_i$ then y_i is defined as a restriction of x_i whereas $w_i = (Q_i - y_i)$ is the partial availability of x_i if $y_i > 0$. Therefore, a picking process is incoherent if $x_i = 0$ or at the attribution of w_i when C does not accept the billing of w_i of PO_j. Table 1 shows an OBS problem by means of a set of hypothetical data, which represents the summarization of *FP* and of the *POP* in t and that is used as the basis for the mathematical formulation, computational experimentation and evaluation of the GA proposed.

Table 1 shows the OBS problem and the set of notations employed to characterize the *FP* and the*POP*, which make it self-explanatory. Please note that in the *POP*, the "Accepts Partial Product" column portrays the parameters that specify whether C_a accepts the billing of w_i of PO_j, which are called $C_\alpha^{w_{ijYes}}$ for "Yes" and $C_\alpha^{w_{ijNo}}$ for "No". The restrictions, parameters and decision variables which configure the OBS problem are listed below:

- Indexes

i : Index which notes the ithRg (products registry) on the *FP*;

j : Index which notes the jth requested on n *POP* requests;

α : Index which notes the ath client referring to jth requested by the *POP*;

Table 1 Stock of products available for sale and list of purchase orders in the portfolio

Stock (*FP*)		Purchase order portfolio (*POP*)							
Product register (Rg_i)	Quantity in stock (x_i)	Order number (PO_j)	Client code (C_a)	Product register (Rg_i)	Quantity ordered (q_{ij})	Unit sales price (pr_{ij})	Total billing (TB_{POP})	Date order was processed (d)	Accepts partial $q_{ij}(w_{ij})$
32417	1	100	10	372300	30	540.00	16,200.00	07/05/2015	Yes
38638	2	100	10	276618	30	464.81	13,944.30	07/05/2015	Yes
98152	4	200	20	1166149	1	6380.65	6380.65	07/05/2015	Yes
98160	5	300	30	372300	30	540.00	16,200.00	07/05/2015	Yes
98830	1	300	30	276618	5	464.81	2324.05	07/05/2015	Yes
137539	2	300	30	1166149	3	6380.65	19,141.95	07/05/2015	Yes
137620	2	300	30	726422	3	338.00	1014.00	07/05/2015	Yes
154517	1	300	30	851337	2	420.34	840.68	07/05/2015	Yes
186106	2	300	30	98830	1	1089.82	1089.82	07/05/2015	Yes
260349	3	400	40	372300	4	540.00	2160.00	07/05/2015	No
276618	5	400	40	726422	6	338.00	2028.00	07/05/2015	No
372300	30	500	50	98152	4	740.52	2962.08	07/05/2015	No
408658	–	500	50	726422	4	338.00	1352.00	07/05/2015	No
580282	1	600	60	98152	1	740.52	740.52	08/05/2015	No
726422	7	600	60	98160	3	624.50	1873.50	08/05/2015	No
851337	2	700	70	98152	3	740.52	2221.56	08/05/2015	Yes
1158608	1	700	70	98160	2	624.50	1249.00	08/05/2015	Yes
1166149	1	800	80	186106	2	624.95	1249.90	09/05/2015	Yes
1169441	3	800	80	408658	1	1272.16	1272.16	09/05/2015	Yes
		900	90	98160	4	624.50	2498.00	10/05/2015	No
		900	90	137620	2	2298.63	4597.25	10/05/2015	No
		1000	100	1166149	1	6380.65	6380.65	10/05/2015	No
Total	–	Total	–	–	–	–	107,720.07	–	–

- Parameters and constraints

t : *FP* situation and the *POP* in a determined moment in time;
PO_j : Refers to jth requested by *POP* on t.
C_a : Refers to ath client of the *POP* on t;
x_i : Amount of Rg_i available on *FP* on t;
q_{ij} : Rg_i demand on PO_j of the *POP* on t;
Q_i : Total q_{ij} demand of the *POP* on t;
y_i : x_i restriction on *FP* on t;
w_i : Partial x_i $(Q_i - y_i)$ availability on *FP* case $y_i > 0$ on t;

- Decision variables

d : PO_j; treatment date
$C_\alpha^{w_{ijYes}}$: Determines that the client accepts w_i income to PO_j;
$C_\alpha^{w_{ijNo}}$: Determines that the client doesn't accept w_i income to PO_j;

Therefore, if q_{ij} and w_{ij} refer to the total and partial attributions of q_i to the PO_j respectively, and if *BO* is the income to be obtained, then, the OBS problem to maximize the *POP* income on a given t moment can be expressed as a BO_{max} programming model:

$$\text{Maximize } BO_{max} = max \sum_{j=1}^{PO_j} \sum_{i=1}^{n} q_{ij}.pr_{ij} + \sum_{j=1}^{PO_j} \sum_{i=1}^{n} w_{ij}.pr_{ij} \tag{1}$$

Subject to :

$$x_i > 0 \qquad i = 1, 2, \ldots, n \tag{2}$$

$$Q_i \leq x_i \qquad i = 1, 2, \ldots, n \tag{3}$$

$$w_{ij} > 0 \text{ if } C_\alpha^{w_{ijYes}} \qquad j = 1, 2, \ldots, m \tag{4}$$

$$PO_j \text{ order for } d \text{ descending} \tag{5}$$

The restriction (2) assures that x_i will only be assigned to PO_j if $x_i > 0$ in the *FP* in t. The restriction (3) guarantees that q_i can only be attributed to PO_n if $Q_i \leq x_i$ in the *FP* in t. The restriction (4) will check if w_i is being attributed to

PO_j case $C_\alpha^{w_{ijYes}}$. The variable (5) will allow that d of PO_j be satisfied. Essentially, the OBS may then be defined as an optimization of the picking process in view of the relevance of x_i for q_i of PO_j in order to satisfy Q_i of POP on the d negotiated, depending on the boundary conditions established by restrictions and on the predefined decision-making criteria, in which the solution expected is the maximization of the billing.

Hybrid Genetic Algorithm

John Holland proposed GAs in 1975. A GA is a global optimization heuristic that uses an initial set of solutions, called a population, where every individual is a candidate to solve the problem (Goldberg 1989; Mitchell 1996; Gen et al. 2008). Genetic Algorithms may be defined as a generic adaptive search method that imitates the genetic process and Darwin's natural evolution of living beings by means of the selection, reproduction and survival of the fit strings, with a structured, despite random, exchange of information (Goldberg 1989; Michalewicz 1996). Such strings or artificial chromosomes represent the individuals, which are formed by genes that quantify possible solutions to a problem (Bäck and Schwefel 1993; Mitchell and Taylor 1999).

The main ideia of GAs is to improve solutions by making evolutive changes to the chromosomes by means of genetic operators (De Jong 1988; Gen and Cheng 2000; Biegler and Grossmann 2004). Initially, based on parameters, the information on the problem is represented in the chromosomes according to the codification schema chosen (Gen and Cheng 1997; Yang 2005). From there onward, n individuals are generated to compose the initial population that usually has a constant size (Whitley 1994; Gen et al. 2008). Next, each individual is evaluated by a fitness function, which measures the quality of the solution to the problem (Holland 1975; Goldberg 1989). Selection elects a percentage of the most fit individuals to be subjected to genetic changes by means of crossover and mutation operators, which are applied according to previously determined crossover (p_c) and mutation (p_m) probabilities (Goldberg 1989; Mitchell 1996).

Reproduction is equivalent to sexual reproduction and mutation aids genetic diversity, thus avoiding premature convergences (Prebys 1999; Gen et al. 2008). The population of the next generation is then formed by the children generated by the previous population and by inserting new individuals that will complete it (Gen et al. 2008). The effectiveness of the members of such population, as a solution to the problem, is evaluated by the fitness function (Michalewicz 1996; Gen and Cheng 2000). Evaluation is responsible for the evolution of the population, since the most fit have better chances of survival and transmitting their genetic material to the next generations (Whitley 1994; Mitchell 1996).

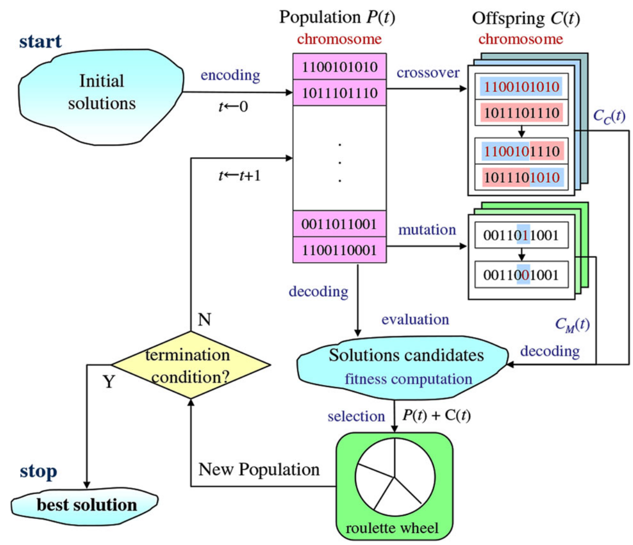

Fig. 1 The general structure of Genetic Algorithms (Gen et al. 2008)

After n generations have occurred, the stopping conditions have to be satisfied and it is expected that the final population will present a solution that, if not excellent, is the best possible for the problem (Goldberg 1989; Michalewicz 1996). Figure 1 shows a general structure of GA. Let $P(t)$ and $C(t)$ be parents and offspring in current generation t, the general implementation structure of GA is described in Fig. 2.

Figure 1 in conjunction with the general structure GA application described in Fig. 2 demonstrates that when a stopping condition is satisfied, the GA stops and provides the best individual as solution. While the stopping condition is not satisfied, the GA executes further interations that consist in applying genetic operations to the current population. Such process generates a new solution that is evaluated in the same manner as the one before and such process is repeated while necessary during the entire execution of the GA.

Thus, the following subsections describes the steps to formulate and implement each element of the GA proposed to solve the OBS expressed by means of mathematical modeling. The conception of GA methodically follows the steps addressed in the bibliographic references and the programming environment used is the Microsoft Office Excel 2010 Visual Basic for Applications (VBA). The next section describes the computational experimentations and performance analysis of the GA in a 2.8 GHz, Core i5 with 8 GB RAM and HD 750 GB.

Maximum possible billing (MB)

The *MB* is the maximum billing amount that can be obtained relating to the set of data existing at t. The purpose of the *MB* is to serve both as a verifier of the need to execute the

Fig. 2 The general implementation structure of Genetic Algorithms (Lin and Gen 2009)

```
procedure: Simple GA
input: problem data, GA parameters
output: a best solution
begin
    t ← 0;                                   // t: generation number
    initialize P(t) by encoding routine;     // P(t): population of chromosomes
    fitness eval(P) by decoding routine;
    while (not termination condition) do
        crossover P(t) to yield C(t);        // C(t): offspring
        mutation P(t) to yield C(t);
        fitness eval(C) by decoding routine;
        select P(t+1) from P(t) and C(t);
        t ← t+1;
    end
    output a best solution;
end
```

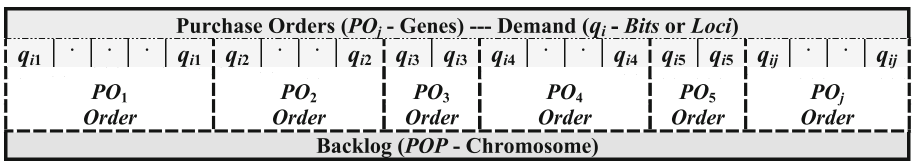

Fig. 3 Picking process chromosome representation

GA and a parameter that improves the picking mechanism. Hence, if w_i is a partial attribution of x_i, and if each i can have a different unit price pr_i for each PO_j, because of the various matters that influence negotiations with C_a then b_{wij} is the billing amount of w_{ij} when q_{ij} is not fully satisfied due to the restrictions of x_i and b_{qij} the billing amount of all i whose x_i is more than enough to satisfy q_{ij}. According to such assumption, the calculation criteria to obtain the *MB* is to prioritize qi as greater than pr_{ij}, i.e., if $pr_{i2} > pr_{ij}$, then the billing of q_{i2} is simulated, and so on, depending on the restriction, as demonstrated by expression 6.

$$\text{if } Q_i > x_i \rightarrow pr_{ij}.w_{ij} = b_{w_{ij}} \text{ or}$$
$$\text{if } Q_i \leq x_i \rightarrow pr_{ij}.q_{ij} = b_{q_{ij}} \tag{6}$$

The calculation of the *MB* can be written based on expression 1 as presented in Eq. 7,

$$MB = \sum_{\substack{j=1 \\ i=1}}^{N_{bits}} b_{q_{ij}} + \sum_{\substack{j=1 \\ i=1}}^{N_{bits}} b_{w_{ij}} \tag{7}$$

By adopting the notation TB_{POP} as the total billing of the *POP*, Eq. 7 verifies the actual need for executing the GA, and that process is performed as follows: if $x_i \geq Q_i$, then $MB = TB_{POP}$, therefore x_i is sufficient to satisfy all PO_j. Otherwise, it is implied that there are y_i restrictions for x_i; hence, it is necessary to execute the GA to find, among the possible alternatives, an optimal solution for the OBS.

Chromosome representation

The manner of representing the picking process in the structure of the string is an adaptation of the binary chromosome representation proposed by Zukhri and Omar (2006) in which every position of the chromosome is a binary vector type solution s, where $s \epsilon \{0, 1\}$. Thus, item i may, or may not, be supplied for the jth purchase order. Therefore, if the binary selection is $s_{ij} = 1$, then $x_i \epsilon PO_j$, which means there has been a picking process, otherwise $s_{ij} = 0$. Hence, *POP* chromosome divides into PO_j genes, and the jth gene represents the jth purchase order, where q_i is an allele. Chromosome representation is illustrated by Fig. 3.

Initial population generation

The initial population is randomly generated by means of a random number generator with uniform distribution. By employing the logical notation proposed by Haupt and Haupt (2004), an N_{pop} matrix represents the population; N_{bits} is the number of bits in the string; and N_{ger} refers to the generations. In the parameter setting, N_{pop} is always defined by the user and never varies from the current N_{ger} to $N_{ger}+1$ during

Table 2 List of purchase orders in the portfolio and initial population generation (N_{pop})

Order number	Client code	Product register	Quantity ordered	Date order was processed	Accepts partial q_{ij}	Unit sales price	Total billing	Chromosomes (C) of the initial N_{pop} of individuals (i)			
PO_j	C_a	Rg_i	q_{ij}	d	w_{ij}	pr_{ij}	TB_{POP}	C_1	C_2	C_3	C_4
Purchase order portfolio (POP)											
100	10	372300	30	07/05/2015	Yes	540.00	16,200.00	1	1	1	1
100	10	276618	30	07/05/2015	Yes	464.81	13,944.30	1	1	1	1
200	20	1166149	1	07/05/2015	Yes	680.65	6380.65	1	1	1	1
300	30	372300	30	07/05/2015	Yes	540.00	16,200.00	1	0	1	0
300	30	276618	5	07/05/2015	Yes	464.81	2324.05	0	1	1	1
300	30	1166149	3	07/05/2015	Yes	6380.65	19,141.95	1	0	0	0
300	30	726422	3	07/05/2015	Yes	338.00	1014.00	0	1	1	1
300	30	851337	2	07/05/2015	Yes	420.34	840.68	0	1	1	0
300	30	98830	1	07/05/2015	Yes	1089.82	1089.82	0	1	1	1
400	40	372300	4	07/05/2015	No	540.00	2160.00	1	1	0	1
400	40	726422	6	07/05/2015	No	338.00	2028.00	1	0	1	0
500	50	98152	4	07/05/2015	No	740.52	2962.08	1	1	1	1
500	50	726422	4	07/05/2015	No	338.00	1352.00	1	0	1	0
600	60	98152	1	08/05/2015	No	740.52	740.52	0	0	1	0
600	60	98160	3	08/05/2015	No	624.50	1873.50	1	1	1	1
700	70	98152	3	08/05/2015	Yes	740.52	2221.56	1	1	1	1
700	70	98160	2	08/05/2015	Yes	624.50	1249.00	0	0	1	0
800	80	186106	2	09/05/2015	Yes	624.95	1249.90	1	0	1	1
800	80	408658	1	09/05/2015	Yes	1272.16	1272.16	1	0	1	0
900	90	98160	4	10/05/2015	No	624.50	2498.00	0	1	0	1
900	90	137620	2	10/05/2015	No	2298.63	4597.25	0	1	1	1
1000	100	1166149	1	10/05/2015	No	6380.65	6380.65	1	1	1	1
Totals	–	–	–	–	–	–	107,720.07	–	–	–	–

Fig. 4 Procedure to the generation of chromosomes of initial population of individuals (N_{pop})

```
procedure: initial population generation
input: GA parameters (population size - Npop), problem data (individual size)
output: initial population
begin
        Npop = empty array;
        for i in population size:
                individual = empty array
                for C in range(Npop = size of POP):
                        individual[C] ← random number between 0,1
                Npop[i] ← individual
        output: initial Npop
end
```

Table 3 List of purchase orders in the portfolio and decoding of chromosomes

Purchase order portfolio (POP)								Decoding of chromosome (C)			
PO_j	C_a	Rg_i	q_{ij}	d	w_{ij}	pr_{ij}	TB_{POP}	C_1	C_2	C_3	C_4
100	10	372300	30	07/05/2015	Yes	540.00	16,200.00	16,200.00	16,200.00	16,200.00	16,200.00
100	10	276618	30	07/05/2015	Yes	464.81	13,944.30	2324.05	2324.05	2324.05	2324.05
200	20	1166149	1	07/05/2015	Yes	6380.65	6380.65	6380.65	6380.65	6380.65	6380.65
300	30	372300	30	07/05/2015	Yes	540.00	16,200.00	16,200.00	–	–	–
300	30	276618	5	07/05/2015	Yes	464.81	2324.05	–	2324.05	–	2324.05
300	30	1166149	3	07/05/2015	Yes	6380.65	19,141.95	19,141.95	–	–	–
300	30	726422	3	07/05/2015	Yes	338.00	1014.00	–	1014.00	1014.00	1014.00
300	30	851337	2	07/05/2015	Yes	420.34	840.68	–	840.68	840.68	–
300	30	98830	1	07/05/2015	Yes	1089.82	1089.82	–	1089.82	1089.82	1089.82
400	40	372300	4	07/05/2015	No	540.00	2160.00	2160.00	2160.00	–	2160.00
400	40	726422	6	07/05/2015	No	338.00	2028.00	2028.00	–	1352.00	–
500	50	98152	4	07/05/2015	No	740.52	2962.08	2962.08	2962.08	2962.08	2962.08
500	50	726422	4	07/05/2015	No	338.00	1352.00	338.00	–	–	–
600	60	98152	1	08/05/2015	No	740.52	740.52	–	–	–	–
600	60	98160	3	08/05/2015	No	624.50	1873.50	1873.50	1873.50	1873.50	1873.50
700	70	98152	3	08/05/2015	Yes	740.52	2221.56	2221.56	2221.56	–	2221.56
700	70	98160	2	08/05/2015	Yes	624.50	1249.00	–	–	1249.00	–
800	80	186106	2	09/05/2015	Yes	624.95	1249.90	1249.00	–	1249.90	1249.90
800	80	408658	1	09/05/2015	Yes	1272.16	1272.16	1272.16	–	–	–
900	90	98160	4	10/05/2015	No	624.50	2498.00	–	1249.00	–	1249.00
900	90	137620	2	10/05/2015	No	2298.63	4597.25	–	4597.25	4597.25	4597.25
1000	100	1166149	1	10/05/2015	No	6380.65	6380.65	6380.65	6380.65	–	6380.65
Billing obtained (BO)							107,720.07	80,731.60	51,617.29	41,132.93	52,026.51
Fitness function ($F_{fitness}$)							–	33,355.28	37,282.03	41,132.93	37,691.25

the entire execution of the GA. Table 2 illustrates an example of the representation of the referred binary chromosome to the generation of a N_{pop} initial of four chromosomes and Fig. 4 shows the pseudo code of the procedure to the generation of N_{pop} aleatory. After the initiating operator is used, *BO* is the name given for the total billing obtained by the chromosome, as demonstrated in Eq. 8.

$$BO = \sum_{i=1}^{N_{bits}} \sum_{j=1}^{N_{bits}} s_{ij}.q_i.pr_{q_{ij}} \qquad \forall s_{ij} = \{0, 1\} \tag{8}$$

Fitness function ($F_{fitness}$)

$F_{fitness}$ evaluates the fitness level of each chromosome as a solution for the OBS and it is applied to all of the N_{ger} to evaluate the relationships of each individual with the rest of N_{pop}. So that the randomization that is intrinsic to the N_{pop} will not to compromise the solution for the OBS nor the evolution of the GA, by virtue of possible generations of invalid bits, the $F_{fitness}$ penalizes unfeasible solutions. Constraint penalties are applied by attributing a weight, which in

```
procedure: Fitness Function (F_fitness)                    // F_fitness: fitness value of the C
input: individual, problem data
output: F_itness value of individual
begin
        for each item (q_i) with bit 1:
            check if it is possible invoice total or partial (C_α^{wijYes}):
                if x_i > 0 and x_i ≥ q_i invoice ← update stock (x_i) and check next item;
                if x_i > 0 and q_i > x_i and C_α^{wijYes} invoice w_i ← update x_i e check next item;
                if x_i = 0 ← penalize and repair item and check next item;
                if x_i > 0 and q_i > x_i e C_α^{wijNo} ← penalize and repair item;
        for each item (q_i) with bit 1 or 0:
            organize the orders by d and check each item:          // d: item's service date (order)
                if d_1 is invoiced before d ← penalize item d_1
        F_itness ← (BO) – total penalties;                        // BO: total billing obtained by C
        se F_itness < 0
                F_itness ← 1,00;
        output: F_itness value of individual;
end
```

Fig. 5 Procedure of penalties applications and calculus of $F_{fitness}$

this case corresponds to the b_{qij} of each violating bit, which directly affects the fitness of the individual and alters its evolutive process. The constraint penalties are the following:

- *The picking process may not take place for a product that does not exist in the FP stock:* If the action prescribed for the occurrence of picking is given by s_{ij}, the penalty referring to lack of stock Pe_s that guarantees the quantity of x_i is not exceeded is presented by Eq. 9.

$$Pe_{s_{ij}} = \sum_{j=i}^{N_{bits}} b_{q_{ij}} \text{ if } s_{ij} = 1 \forall x_i = 0 \rightarrow pr_{ij}.q_{ij} = b_{q_{ij}} \tag{9}$$

- *The picking process may not take place for w_{ij} when not accepted by the client:* The occurrence of such event implies in the penalty called Pe_w and it is applied when $x_i < Q_i$ and the variable C assumes $C_\alpha^{w_{ijNo}}$ for w_{ij}. The Pe_w is presented by Eq. 10.

$$Pe_{w_{ij}} = \sum_{j=i}^{N_{bits}} b_{q_{ij}} \text{if } s_{ij} = 1 \forall w_{ij} \leftrightarrow C_a^{w_{ijNo}} \rightarrow pr_{q_{ij}}.q_{ji} = b_{q_{ij}} \tag{10}$$

The penalties described above only apply once to the b_{qij} of each bit, regardless of whether, or not, the bit in question has violated more than one criterion liable to punishment. Such penalties are applied according to Eq. 11.

```
procedure: reparation of individual
input: item, individual, problem data
output: none
begin
        order ← get order from item.
        for each item in order:
                item value ← 0
                item quantity ← 0
                chromosome in individual related to item ← 0
                update stock
end
```

Fig. 6 Reparation process of individual

$$Pe_{s_{ij}} = \begin{cases} b_{q_{ij}} & \text{if } s_{ij} = 1 \forall x_i = 0 \\ 0 & \text{if } s_{ij} = 0 \end{cases}$$
$$Pe_{w_{ij}} = \begin{cases} b_{q_{ij}} & \text{if } Pe_s = 0 \\ b_{q_{ij}} & \text{if } s_{ij} = 1 \forall w_{ij} \leftrightarrow C_a^{w_{ijNo}} \\ 0 & \text{if } s_{ij} = 0 \end{cases} \tag{11}$$

Then, the GA begins the verification of the attributions according to the d criterion. In cases where $x_i < Q_i$, the PO_j referring to the d_1 have billing preference in relation to d_2. Thus, in order to make it possible to direct the search process, the GA applies a penalty Pe_d that, contrary to the previous ones, occurs after the selection operator is used and may be applied to a bit that has already been punished, according to Eq. 12.

$$\begin{aligned} Pe_{d_{ij}} &= \sum_{j=i}^{N_{bits}} f_{q_{ij}} \text{ if } x_i < Q_i \leftrightarrow s_{ij} \\ &= 1 \forall P_j^{d+1} \wedge s_{ij} = 0 \forall P_j^d \rightarrow pr_{ij}^{d+1}.q_{ji} = b_{q_{ij}} \end{aligned} \tag{12}$$

Individual (*String*) C_i	Fitness Function $F_{fitness_i}$	Selection Percentage (p_i)	Piece of Roulette (°)
C_1	**33,355.28**	22.32%	80.34
C_2	**37,282.03**	24.94%	89.80
C_3	**41,132.93**	27.52%	99.07
C_4	**37,691.25**	25.22%	90.78
Total	**149,461.49**	**100.00%**	**360.00**

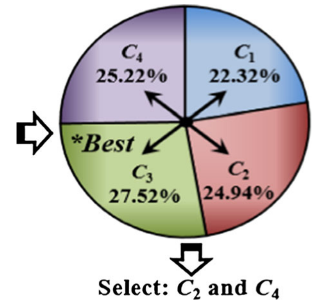

Fig. 7 Calculus of evaluation and select graphic by the roulette wheel with elitism

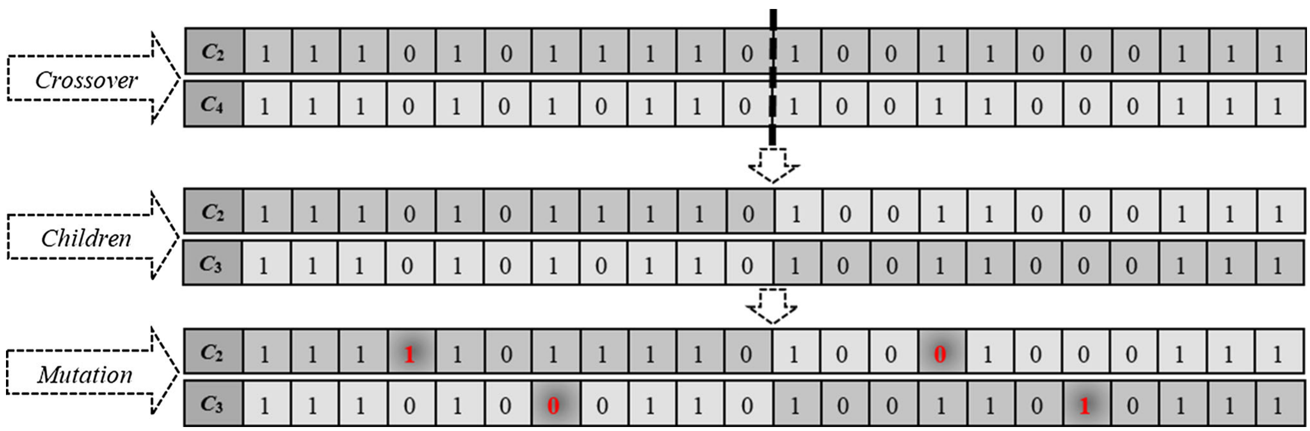

Fig. 8 Diagram of crossover and mutation

Evidently, bits = 1 suffer double penalties under certain circumstances, which might result in a negative *BO* value. In this case, \$1.00 is attributed to *BO* for $BO \leq 0$ to avoid problems with the selection operator. Taking *BO* as the basis to evaluate the chromosome, the $F_{fitness}$ that evaluates the picking process is presented by Eq. 13.

$$F_{fitness} = \begin{cases} BO - \sum_{j=1}^{N_{bits}} (Pe_{s_{ij}} + Pe_{w_{ij}} + Pe_{d_{ij}}) \\ \text{if } BO - \sum_{j=1}^{N_{bits}} (Pe_{s_{ij}} + Pe_{w_{ij}} + Pe_{d_{ij}}) \\ \qquad < 1 \rightarrow F_{fitness} = \$1.00 \end{cases} \tag{13}$$

According to the $F_{fitness}$, the fittest individuals are those that obtain a higher *BO* after the penalties are applied. Please note that the penalties are corrective and have the objective of helping the N_{pop} to evolve, since the fittest individuals have more changes of transferring their genetic material to the $N_{ger} + 1$. In short, the value of the $F_{fitness}$, presupposing that it may vary from 1 to the *BO*, quantitatively determines whether such individual satisfies, or not, the conditions imposed by the problem and contributes to make the maximization of the billing the optimized result of the OBS. After the calculation of $F_{fitness}$ the reparation process of individuals starts which, based in the order number, attributes 0 to the invalid bits of a chromosome. Therefore, the bits = 1 which refer to the picking process when $x_i = 0$ and of w_i when $C_\alpha^{w_{ijNo}}$ will be substituted by bits = 0. In concomitance the respective x_i will be updated going back to the quantity of items attributed improperly to the stock. Table 3 illustrates the decryption of four chromosomes and so to a better visualization the following colors are used to differentiate the N_{bits}; i) *bits* in color black refer to attribution of q_{ij} *bits* in color green to w_{ij} and; ii) the color red indicates the bits penalized by attributions of q_{ij} and the color blue the bits penalized by the attributions of w_{ij}. In the sequence, Fig. 5 shows the procedures of penalties applications and calculus of $F_{fitness}$ and Fig. 6 the reparation process of individuals.

Selection operator

The selection method adopted is the fitness proportionate selection, also known as the roulette wheel selection with elitism, proposed by Holland (1975), where p_i is the probability of selection of each individual i equivalent to a certain slice of the roulette wheel. If the evaluation of N_{pop} is proportional to the evaluation of $F_{fitness}$, the solutions obtaining a higher $F_{fitness}$ as compared to the others have a higher p_i, or have more chances of being chosen as a solution for the OBS for they represent larger slices of the roulette wheel. With regard to the quantity of N_{pop} selected to participate in the crossover and mutation, we have fixed it at 50 % ($N_{pop}/2$) of the N_{pop}. Therefore, if the quotient of $N_{pop}/2$ is an odd number, then $N_{pop}/2+1$ individuals are selected to form the crossover pairs to be ordered by the $F_{fitness}$. Given the N_{pop} and the $F_{fitness}$, the p_i is defined by Eq. 14 and Fig. 7 illustrates the roulette wheel graphic with elitism to the example of Table 3.

$$p_i = \frac{F_{fitness_i}}{\sum_{i=1}^{N_{pop}} F_{fitness_i}} \tag{14}$$

In the referred selection method only the best individual of each N_{ger}, which in the example of Fig. 7 is C_3, will be transferred integrally to become the first individual of $N_{ger}+1$. Therefore, the risk of this individual not being selected or even destroyed by crossover and mutation is eliminated. However, after the spin, the roulette will stop in an random point determined by the chromosome to be selected, and this process is repeated as many times as necessary until you have the sufficient number of pairs to the application of a new crossover.

Crossover and mutation operators

The crossover operator of a point (Holland 1975), where the cut-off position is set at 50 % of the N_{bits}, executes the breeding between the pairs of all of the chromosomes selected as parents. The mutation operator is applied to all of the children resulting from the crossover using the random bit exchange technique in which the user has the option of informing the p_m wanted, which may range from 0 to 100 %, according to the need for evolution in the solutions. Figure 8 illustrates the crossover diagram and mutation to the formation of chromosomes *children* of $N_{ger}+1$. Take notice that the example on Fig. 8 that C_2 and C_4 will become C_2 and C_3 and will join to the chromosome C_1 (elitism) in $N_{ger}+1$, while C_4 will be generated by the technique of the new population generation.

Generation of the new population

The elitist selection introduced by De Jong (1988) is applied to the N_{pop}. In this case, only the fittest individual of each cycle will be fully transferred to $N_{ger}+1$, which eliminates the risk of such individuals not be selected or even destroyed by the crossover and mutation operators. Thus, it is guaranteed that the *BO* obtained at each N_{ger} is at least equal or higher than the *BO* of $N_{ger}-1$. However, it is important to highlight that, as defined in "Fitness function ($F_{fitness}$)" section, the $F_{fitness}$ of the fittest individual may diminish from N_{ger} to $N_{ger}+1$ because of the application of a Pe_d. Therefore, in order to generate the new N_{pop} it has to be considered that the N_{pop} is always fixed (please refer to "Fitness function ($F_{fitness}$)" section) and that the roulette wheel selects 50 or 50 % +1 chromosomes (please refer to

Table 4 Statistical results of the first experimentation

Sequence of executions	Population (N_{pop})	Generations (N_{ger})	Individuals selected	Individuals transferred	Total crossovers	Rate of mutation (%)	Total mutations	N_{pop} fittest individual	Fittest individual	Billing ($F_{fitness}$)	Time minutes
1	10	50	300	49	150	5	300	39	3	40,043.11	00:03:34
2	10	50	300	49	150	5	300	24	10	40,043.11	00:03:48
3	10	50	300	49	150	5	300	44	3	39,042.35	00:03:31
4	10	50	300	49	150	5	300	38	8	40,043.11	00:03:55
Average	10	50	300	49	150	5	300	36	6	39,792.92	00:03:42

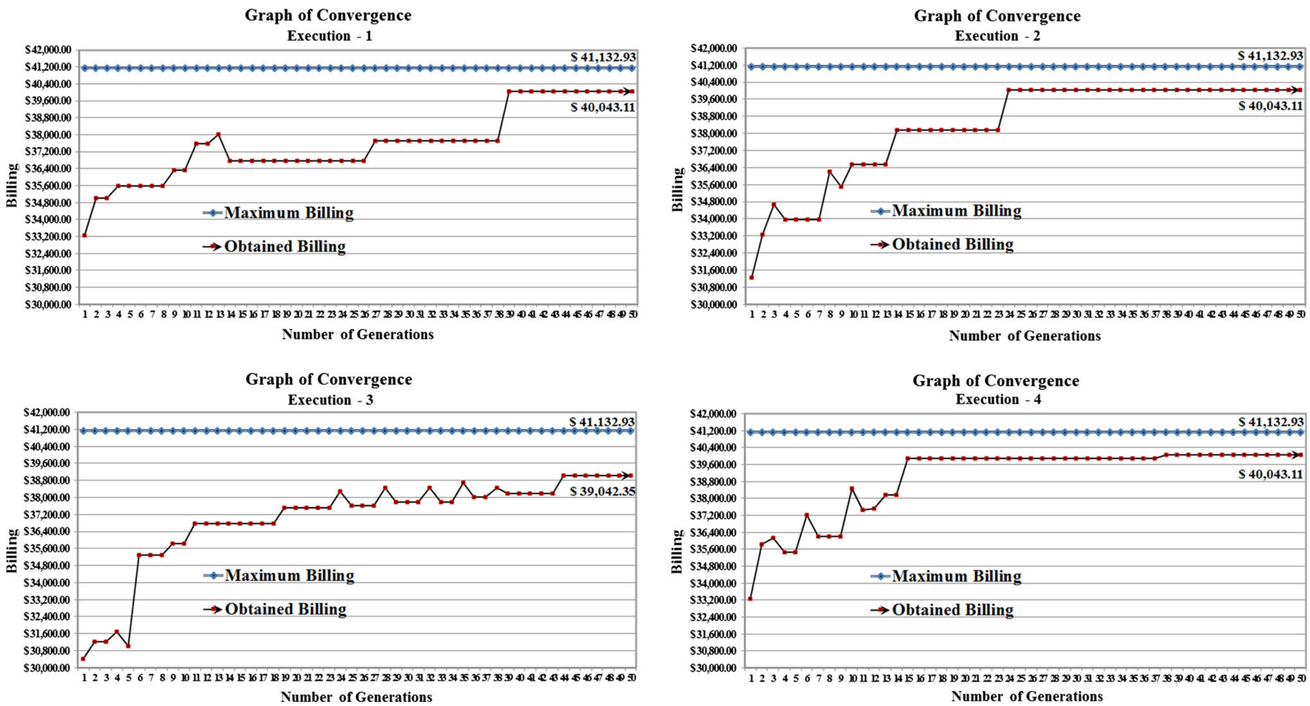

Fig. 9 Graphs on the convergence of the first experimentation—$N_{pop} = 10$, with a 5 % mutation rate

"Selection operator" section). Therefore, if only the fittest individual of the current N_{pop} is transferred to $N_{ger} + 1$, then $N_{pop} - (N_{pop}/2+1)$ ou $N_{pop} - (N_{pop}/2+2)$ is established as the number of individuals that complete the new N_{pop} generated by the same method used in "Initial population generation" section and repeated in all of the cycles for each execution of the GA.

Stopping criterion

The GA proposed allows users to specify the N_{ger} wanted to terminate the evolution process and the stopping criterion of the GA. In this case, the fact of reaching the maximum value of the fitness function (*MB*) is also adopted as a stopping criterion. Regardless of the number of cycles, if the chromosome generated satisfies the objective required finding *MB* in any one N_{ger}, the execution is terminated, otherwise the GA performs all of the cycles until reaching the N_{ger} specified by the user.

Experimentation and discussion of the results

This section demonstrates the experiments of the GA proposed and the discussion of the results obtained for the problem shown in Table 1. The performance assessments were performed in relation to the average results obtained in the various experiments and per set of results reached. In addition to evaluating the level of evolution during all of the N_{ger}, we also evaluated the number of times in which the GA reached the *MB*. We performed 4 experiments where the initial N_{pop} ranged from 10 to 30 individuals, according to the performance and convergence of the GA. All of the experiments simulated 50 N_{ger} at each execution, employing a mutation rate ranging from 5 to 10 % where the convergence graphs referring to the $F_{fitness}$ of the fittest individual of each N_{ger} are demonstrated. Table 4 presents the results of the first experiment and Fig. 9 shows the level of convergence of the GA.

As illustrated in Fig. 9, the number of individuals used is sufficient for GA convergence. The results of this study show that up to a certain N_{ger} number the GA reached certain levels of evolution. However, results do not improve after a certain point. Logo, Hence, because of elitism, the best solution of each N_{ger} repeats itself in $N_{ger} + 1$. Please note that after a certain point, there are situations of regression in the level of evolution of the billing due to the action of the Pe_d, which punishes even the fittest individuals that violate the conditions of the d imposed by the OBS. Therefore, the occurrence of such events is common, which can also be observed in other graphs of all of the experimentations. In the next experiment, shown in Table 5 and illustrated in Fig. 10, 16 individuals are used at an 8 % mutation rate.

In this second set of simulations, we noted that performance was improved with a small increase in the number of individuals. Figure 10 shows that the GA found the *MB* at

Table 5 Statistical results of the second experimentation

Sequence of executions	Population (N_{pop})	Generations (N_{ger})	Individuals selected	Individuals transferred	Total crossovers	Rate of mutation (%)	Total mutations	N_{pop} fittest individual	Fittest individual	Billing ($F_{fitness}$)	Time minutes
1	16	47	376	46	188	8	752	47	16	41,132.93	00:03:37
2	16	50	400	49	200	8	784	47	12	40,118.93	00:07:50
3	16	50	400	49	200	8	784	42	6	40,043.11	00:13:54
4	16	36	288	35	144	8	576	36	7	41,132.93	00:12:32
Average	16	46	366	45	183	8	724	43	10	40,606.98	00:09:28

the first and fourth executions. However, the average result obtained is not satisfactory since the desirable result was achieved in only 50 % of the executions and there is a risk the GA will be restricted to a local solution for other executions. Obviously, the more the N_{pop} is increased the greater the probability of finding better results. Nevertheless, there is also a considerable increase in computational time that tripled in average reaching 09:28 min. Twenty individuals and a 10 % mutation rate were used in the third experiment as shown by Table 6 and Fig. 11.

In the third experiment, we can see that with an increase in the N_{pop} the level of evolution of the offspring increases considerably and the GA finds the *MB* in all of the executions. Such a scenario is clearly reflected in the graphs of Fig. 11 showing that despite the contribution of mutation genetic variety is essentially obtained by population increase, which makes it easier for the GA to find better solutions. It is important to explain that, in spite of the fact that the GA found *MB* at generation number 35 on the third execution, it does not respect the date criterion. Hence, the result is not considered and the process continues up to generation 41 where an individual that is fit for the OBS is originated. Thirty individuals and a 10 % mutation rate were used in the fourth experiment presented by Table 7 and Fig. 12.

Upon analysing the results of the simulations for 30 individuals (Fig. 12), we found that the frequency of convergence to *MB* was quick, 10 N_{ger}, within 06:07 min in average. This means that the size of the N_{pop} was sufficient to produce a diversity level capable of representing all of the search space of the problem, i.e., it is from such environment onward that the best performance of the GA is verified. The combined evaluation of the results achieved provides an overall understanding of the behaviour of the GA with respect to the ideal size of the N_{pop} and the complexity of the analyses. It is unquestionable that, in OBS situation where the managers need to make decisions quickly, the reaching the maximum value of the $F_{fitness}$ depends on the ideal size of the N_{pop}. Therefore, if the size of the N_{pop} is small the crossover and mutation effects are also small and there will not be genetic variety for the evolution of the GA, i.e., the higher the *POP* and the complexity of the parameters the greater the number of individuals needed for the GA to work well.

Final considerations

This article presented the formulation of a GA to optimize a specific OBS problem. The GA was applied to a set of data represented by *FP* and *POP* and was configured to satisfy predefined decision-making restrictions and parameters. In the experiments conducted, both the results obtained and the computational processing time were satisfactory and coherent with regard to the scope of the research. As demon-

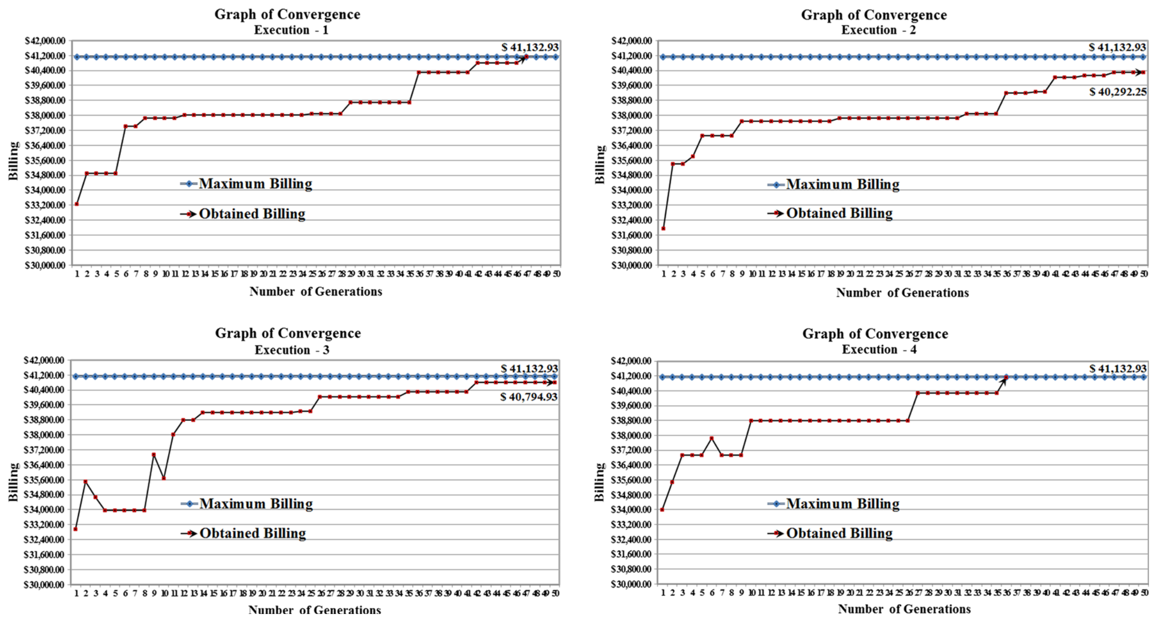

Fig. 10 Graphs on the convergence of the second experimentation—$N_{pop} = 16$, with an 8 % mutation rate

Table 6 Statistical results of the third experimentation

Sequence of executions	Population (N_{pop})	Generations (N_{ger})	Individuals selected	Individuals transferred	Total crossovers	Rate of mutation (%)	Total mutations	N_{pop} fittest individual	Fittest individual	Billing ($F_{fitness}$)	Time minutes
1	20	29	290	28	145	10	580	29	10	41,132.93	00:02:19
2	20	45	450	44	225	10	900	45	13	41,132.93	00:04:16
3	20	41	410	40	205	10	820	41	10	41,132.93	00:03:13
4	20	34	340	33	170	10	680	34	20	41,132.93	00:03:12
Average	20	37	373	36	186	10	745	37	13	41,132.93	00:03:15

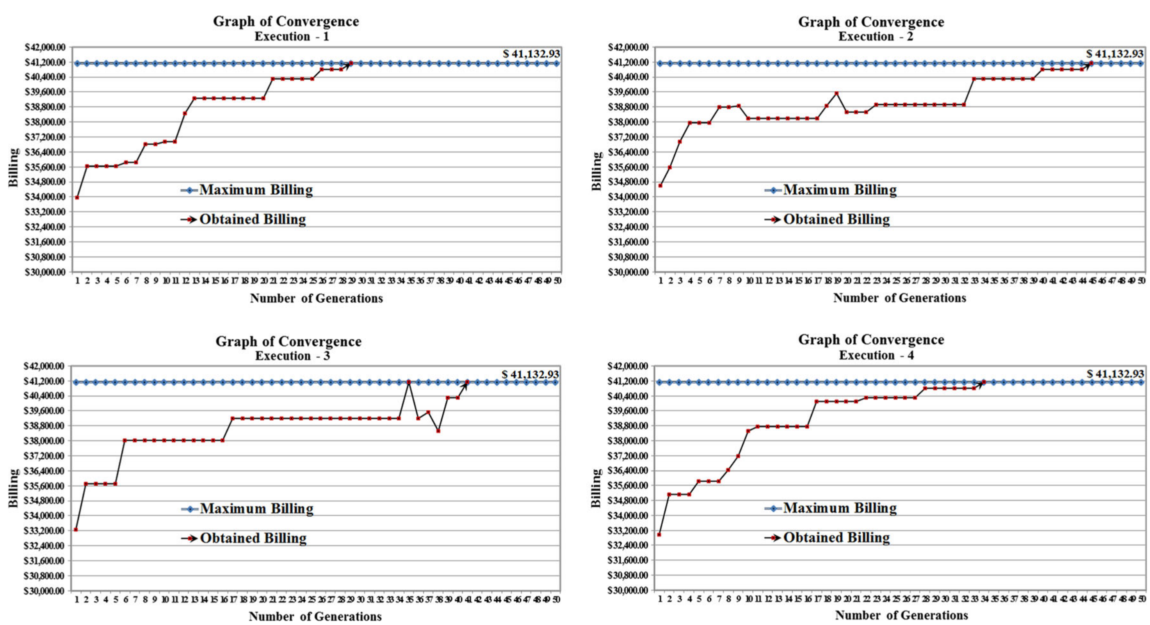

Fig. 11 Graphs on the convergence of the third experimentation—$N_{pop} = 20$, with a 10 % mutation rate

Table 7 Statistical results of the fourth experimentation

Sequence of executions	Population (N_{pop})	Generations (N_{ger})	Individuals selected	Individuals transferred	Total crossovers	Rate of mutation (%)	Total mutations	N_{pop} fittest individual	Fittest individual	Billing ($F_{fitness}$)	Time minutes
1	30	6	96	5	48	10	192	6	8	41,132.93	00:07:54
2	30	7	112	6	56	10	224	7	26	41,132.93	00:05:34
3	30	12	192	11	96	10	384	12	24	41,132.93	00:07:18
4	30	13	208	12	104	10	416	13	12	41,132.93	00:03:40
Average	30	10	152	9	76	10	304	10	18	41,132.93	00:06:07

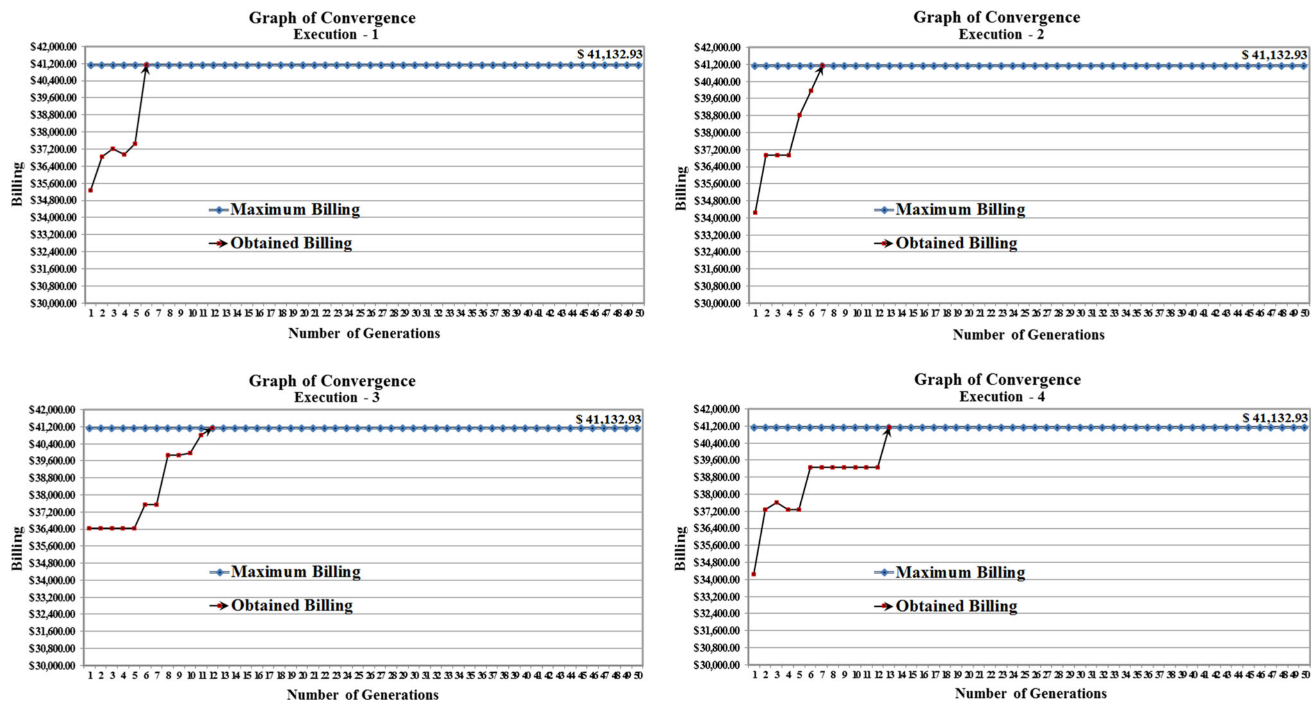

Fig. 12 Graphs on the convergence of the fourth experimentation—$N_{pop} = 30$, with a 10 % mutation rate

strated herein, the GA proposed has an optimization potential capable of making the picking process faster and of maximizing billing. Based on the result obtained, the logistics information flow becomes quicker in identifying which items must be set aside for each client. The production area has a general view of the *FP* as compared to the *POP*, which provides more rapidity and accuracy to programming the productive process. The cost variables of different departments associated to the planning and operating issues of the OBS can be minimized. In short, the solution to the OBS minimizes actual problems and reduces costs, besides providing flexibility conditions and adaptation to changes that, in a certain manner, help achieve effectiveness in a series of processes that involve time, planning, negotiating, and decision-making. Despite the fact that other OBS algorithms were not found for comparison purposes, we have reached the conclusion that the GA proposed is a feasible option for entities that wish to maximize billing and face the type of problem addressed herein. Evidently, further studies on the current topic may contribute with new methods for problems of such nature. A few suggestions for future research include:

- Trying to increase the performance of the GA through other parameters, operators or representations. In this research, for example, we consider a change in mutation rate ranging from 5 to 10 %, but without considering the changed rate of fitness value in each generation. We recommend that in future research these issues should be considered and suggest Yun and Gen (2003) and Lin and Gen (2009) as papers references on parameter tuning by Fuzzy logic controller;
- Conduct studies with actual application to be able to compare more effectively the existing processes with those resulting from the application of the GA. Besides that, to execute a bigger experimentations number and add statistical analysis to the experimental demonstration as, for

example, an ANOVA (variance analysis) to demonstrate the effectiveness of the GA;

- Assess the OBS with respect to the profit margin and dynamic operating issues such as production lead time and work-in-process, thus increasing the level of adaptation in response to ever changing environments;
- Minimize the distance to be run when physically separating the products indicated by the picking process in large warehouses.

References

Ayala, F. J. (2010). Darwin's explanation of design: From natural theology to natural selection. *Infection, Genetics and Evolution*, *10*(6), 840–848.

Bäck, T., & Schwefel, H. P. (1993). An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, *1*(1), 1–28.

Bandyopadhyay, S., & Bhattacharya, R. (2014). Solving a tri-objective supply chain problem with modified NSGA-II algorithm. *Journal of Manufacturing Systems*, *33*(1), 41–50.

Baud-Lavigne, B., Bassetto, S., & Agard, B. (2014). A method for a robust optimization of joint product and supply chain design. *Journal of Intelligent Manufacturing*, pp. 1-9. doi:10.1007/s10845-014-0908-5.

Bertrand, J. W. M., & Fransoo, J. C. (2002). Modelling and simulation: Operations management research methodologies using quantitative modeling. *International Journal of Operations and Production Management*, *22*(2), 241–264.

Biegler, L. T., & Grossmann, I. E. (2004). Retrospective on optimization. *Computers and Chemical Engineering*, *28*(8), 1169–1192.

Blickle, T. (1996). Theory of evolutionary algorithms and application to system synthesis (PhD thesis, Swiss Federal Institute of Technology), TIK-Schriftenreihe Nr. 17. Zurich, Switzerland: vdf Hochschul Verlag GA an der ETH.

Bowler, P. J. (2001). Evolution, history of. In N. J. Smelser & P. B. Baltes (Eds.), *International encyclopedia of the social and behavioral sciences* (pp. 4986–4992). Oxford, UK: Pergamon Press.

Chen, T. L., Chen, Y. Y., & Lu, H. C. (2013). A capacity allocation and expansion model for TFT-LCD multi-site manufacturing. *Journal of Intelligent Manufacturing*, *24*(4), 847–872.

Chien, C., Kim, K. H., Liu, B., & Gen, M. (2012). Advanced decision and intelligence technologies for manufacturing and logistics. *Journal of Intelligent Manufacturing*, *23*(6), 2133–2135.

De Jong, K. (1988). Learning with genetic algorithms: An overview. *Machine Learning*, *8*(2–8), 121–188.

Diabat, A., Abdallah, T., & Henschel, A. (2015). A closed-loop location-inventory problem with spare parts consideration. *Computers and Operations Research*, *54*, 245–256.

Elsayed, S. M., Sarker, R. A., & Essam, D. L. (2014). A new genetic algorithm for solving optimization problems. *Engineering Applications of Artificial Intelligence*, *27*, 57–69.

Gen, M., & Cheng, R. (1997). *Genetic algorithms and engineering design*. New York: Wiley.

Gen, M., & Cheng, R. (2000). *Genetic algorithms and engineering optimization*. New York: Wiley.

Gen, M., Cheng, R., & Lin, L. (2008). *Network models and optimization: Multiobjective genetic algorithms approach*. London: Springer.

Ghiami, Y., Williams, T., & Wu, Y. (2013). A two-echelon inventory model for a deteriorating item with stock-dependent demand, partial backlogging and capacity constraints. *European Journal of Operational Research*, *231*(3), 587–597.

Ghiselin, M. T. (2009). Darwin and the evolutionary foundations of society. *Journal of Economic Behavior and Organization*, *71*(1), 4–9.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning* (2nd ed.). Massachusetts: Addison-Wesley Co.

Gu, J., Goetschalckx, M., & McGinnis, L. F. (2007). Research on warehouse operation: A comprehensive review. *European Journal of Operational Research*, *177*(1), 1–21.

Haegeman, K., Marinelli, E., Scapolo, F., Ricci, A., & Sokolov, A. (2013). Quantitative and qualitative approaches in future-oriented technology analysis (FTA): From combination to integration? Technological Forecasting and Social Change, 80(3), 386–397.

Hallam, J. W., Akman, O., & Akman, F. (2010). Genetic algorithms with shrinking population size. *Computational Statistics*, *25*(4), 691–705.

Haq, A. N., & Boddu, V. (2014). Analysis of enablers for the implementation of leagile supply chain management using an integrated fuzzy QFD approach. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-014-0957-9.

Haupt, Randy L., & Haupt, Sue E. (2004). *Practical genetic algorithms* (2nd ed.). new york: Wiley.

Hiremath, N. C., Sahu, S., & Tiwari, M. K. (2013). Multi objective outbound logistics network design for a manufacturing supply chain. *Journal of Intelligent Manufacturing*, *24*(6), 1071–1084.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press.

İnkaya, T., & Akansel, M. (2015). Coordinated scheduling of the transfer lots in an assembly-type supply chain: A genetic algorithm approach. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-015-1041-9.

Jeang, A. (2015). Robust product design and process planning in using process capability analysis. *Journal of Intelligent Manufacturing*, *26*(3), 459–470.

Kumar, R. S., Tiwari, M., & Goswami, A. (2014). Two-echelon fuzzy stochastic supply chain for the manufacturer-buyer integrated production-inventory system. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-014-0921-8.

Lin, L., & Gen, M. (2009). Auto-tuning strategy for evolutionary algorithms: Balancing between exploration and exploitation. *Soft Computing*, *13*(2), 157–168.

Matthews, J., & Visagie, S. (2013). Order sequencing on a unidirectional cyclical picking line. *European Journal of Operational Research*, *231*(1), 79–87.

McCall, J. (2005). Genetic algorithms for modelling and optimisation. *Journal of Computational and Applied Mathematics*, *184*(1), 205–222.

Michalewicz, Z. (1996). *Genetic algorithms + data structures = evolution programs* (8th ed.). New York: Springer.

Mitchell, M. (1996). *An introduction to genetic algorithms*. Cambridge: MIT Press.

Mitchell, M., & Taylor, C. E. (1999). Evolutionary computation: An overview. *Annual Review of Ecology and Systematics*, *80*, 598–616.

Mousavi, S. M., Bahreininejad, A., Musa, S. N., & Yusof, F. (2014). A modified particle swarm optimization for solving the integrated location and inventory control problems in a two-echelon supply chain network. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-014-0970-z.

Mousavi, S. M., Hajipour, V., Niaki, S. T. A., & Alikar, N. (2013). Optimizingmulti-itemmulti-period inventory control system withdiscounted cash flow and inflation: Two calibrated meta-heuristicalgorithms. *Applied Mathematical Modelling*, *37*(4), 2241–2256.

Nobakhti, A. (2010). On natural based optimization. *Cognitive Computation*, *2*(2), 97–119.

Pando, V., San-José, L. A., & Sicilia, J. (2012). Maximizing profits in an inventory model with both demand rate and holding cost per unit time dependent on the stock level. *Computers and Industrial Engineering*, *62*(2), 599–608.

Park, K., & Kyung, G. (2014). Optimization of total inventory cost and order fill rate in a supply chain using PSO. *The International Journal of Advanced Manufacturing Technology*, *70*(9–12), 1533–1541.

Phillips, F., & Su, Y. S. (2009). Advances in evolution and genetics: Implications for technology strategy. *Technological Forecasting and Social Change*, *76*(5), 597–607.

Prebys, E. K. (1999). The genetic algorithm in computer science. *MIT Undergraduate Journal of Mathematics*, *1*, 165–170.

Richards, G. (2011). *Warehouse management: A complete guide to improving efficiency and minimizing costs in the modern warehouse*. London, UK: Kogan Page Publishers.

Rim, S. C., & Park, I. S. (2008). Order picking plan to maximize the order fill rate. *Computers and Industrial Engineering*, *55*(3), 557–566.

Risdiyono, & Koomsap, P. (2013). Design by customer: Concept and applications. *Journal of Intelligent Manufacturing*, *24*(2), 295–311.

Runarsson, T. P., & Jonsson, M. T. (1999). Genetic production systems for intelligent problem solving. *Journal of Intelligent Manufacturing*, *10*(2), 181–186.

Sadeghi, J., Sadeghi, S., & Niaki, S. T. A. (2014). A hybrid vendor managed inventory and redundancy allocation optimization problem in supply chain management: An NSGA-II with tuned parameters. *Computers and Operations Research*, *41*, 53–64.

Sereshti, N., & Bijari, M. (2013). Profit maximization in simultaneous lot-sizing and scheduling problem. *Applied Mathematical Modelling*, *37*(23), 9516–9523.

Shahabi, M., Akbarinasaji, S., Unnikrishnan, A., & James, R. (2013). Integrated inventory control and facility location decisions in a multi-echelon supply chain networkwith hubs. *Networks and Spatial Economics*, *13*(4), 497–514.

Shen, Z. J. M. (2006). A profit-maximizing supply chain network design model with demand choice flexibility. *Operations Research Letters*, *34*(6), 673–682.

Simchi-levi, D., Kaminsky, P., & Simchi-levi, E. (2003). *Designing and managing the supply chain: Concepts, strategies, and case studies* (2nd ed.). Boston, MA: McGraw-Hill.

Slotnick, S. A. (2011). Order acceptance and scheduling: A taxonomy and review. European Journal of Operational Research, 212(1), 1–11.

Tompkins, J. A., & Smith, J. D. (Eds.). (1998). *The warehouse management handbook*. Raleigh, NC: Tompkins Press.

Triki, H., Mellouli, A., & Masmoudi, F. (2014). A multi-objective genetic algorithm for assembly line resource assignment and balancing problem of type 2 (ALRABP-2). *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-014-0984-6.

Van den Berg, J. P., & Zijm, W. H. M. (1999). Models for warehouse management: Classification and examples. *International Journal of Production Economics*, *59*(1), 519–528.

Wang, C., & Dargahi, F. (2013). Service customization under capacity constraints: An auction-based model. *Journal of Intelligent Manufacturing*, *24*(5), 1033–1045.

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing*, *4*(2), 65–85.

Whitley, D., & Sutton, A. M. (2012). Genetic Algorithms—A Survey of Models and Methods. In *Handbook of Natural Computing* (pp. 637–671). Springer, Berlin Heidelberg.

Xiang, W., Song, F., & Ye, F. (2014). Order allocation for multiple supply-demand networks within a cluster. *Journal of Intelligent Manufacturing*, *25*(6), 1367–1376.

Yang, X. S. (2005). Biology-derived algorithms in engineering optimization. In S. Olarius & A. Y. Zomaya (Eds.), *Handbook of bioinspired algorithms and applications (Chapter 32)* (pp. 585–595). London, UK: Chapman & Hall/CRC Press.

Yang, X. S., & Koziel, S. (2010). Computational optimization, modelling and simulation—A paradigm shift. *Procedia Computer Science*, *1*(1), 1291–1294.

Yao, M. J., & Huang, J. Y. (2014). Optimal lot-sizing and joint replenishment strategy under a piecewise linear warehousing cost structure. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-014-1027-z.

Yun, Y. S., & Gen, M. (2003). Performance analysis of adaptive genetic algorithms with fuzzy logic and heuristics. *Fuzzy Optimization Decision Making*, *2*(2), 161–175.

Zhang, J., Chen, W. N., Zhan, Z. H., Yu, W. J., Li, Y. L., Chen, N., et al. (2012). A survey on algorithm adaptation in evolutionary computation. *Frontiers of Electrical and Electronic Engineering*, *7*(1), 16–31.

Zhuang, Z. Y., & Chang, S. C. (2015). Deciding product mix based on time-driven activity-based costing by mixed integer programming. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-014-1032-2.

Zukhri, Z., & Omar, K. (2006). Implementation of genetic algorithms to cluster new students into their classes. In *8th Seminar Nasional Aplikasi Teknologi Informasi - SNATI 2006* (pp. 101–104). Yogyakarta, Indonesia.