Bi-objective optimization of a multihead weighing process

J. Carlos García-Díaz*

Centre for Quality and Change Management,
Universitat Politècnica de València,
Camino de Vera, s/n. 46022, Valencia, Spain
E-mail: juagardi@eio.upv.es
*****Corresponding author

Alexander Pulido-Rojano

Department of Industrial Engineering,
Simón Bolívar University,
Av. 59 No. 59-92. A.A. 50595, Barranquilla, Colombia
E-mail: apulido3@unisimonbolivar.edu.co

Vicent Giner-Bosch

Centre for Quality and Change Management,
Universitat Politècnica de València,
Camino de Vera, s/n. 46022, Valencia, Spain
E-mail: vigibos@eio.upv.es

Abstract: A multihead weighing process is a packaging technology that can be of strategic importance to a company, as it can be a key to competitive advantage in the modern food industry. The improvement in the process quality and sensory quality of food packaged in a multihead weighing process investigated in this paper is relevant to industrial engineering. A bi-objective ad hoc algorithm based on explicit enumeration for the packaging processes in multihead weighers with an unequal supply of the product to the weighing hoppers is developed. The algorithm uses an a priori strategy to generate Pareto-optimal solutions and select a subset of hoppers from the set of available ones in each packing operation. The relative importance of both aforementioned objectives is dynamically managed and adjusted. The numerical experiments are provided to illustrate the performance of the proposed algorithm and find the optimum operational conditions for the process.

Keywords: Multihead weighing process; Bi-objective optimization; Compromise programming; Exhaustive search; Reduction of variability.

Reference to this paper should be made as follows: García-Díaz, J.C., Pulido-Rojano, A. and Giner-Bosch, V. (xxxx) 'Bi-objective optimization of a multihead weighing process', *European J. Industrial Engineering*, Vol. X, No. Y, pp.

Biographical notes: J. Carlos García-Díaz received his BS in 1991, MS in 1996 and his PhD in Applied Statistics in 2003 from the Polytechnic University of Valencia, Spain. Currently, he is a senior quality consultant and is an Associate Professor of statistical quality control at the Polytechnic University of Valencia.

Author

Alexander Pulido-Rojano, received his B.S. Eng in Industrial Engineering in 2006 from the Universidad de la Costa in Barranquilla (Colombia), his M.S. degree in Industrial Engineering in 2012 from the Universidad del Norte (Colombia), and he is currently undertaking his Doctoral program in Statistics and Optimization, at the Universitat Politècnica de València. Valencia, Spain.

Vicent Giner-Bosch holds a BSc in Mathematics from the Universitat de València, Spain, and a PhD in Applied Statistics, Operations Research and Quality from the Universitat Politècnica de València, Spain. He has been a Faculty member of the Universitat Politècnica de València since 2001.

1 Introduction

1.1 Multihead weighers

Multihead weighers or combinational weighers are used to provide accurate weights at high packing speed, and are they currently the most widely used dosing method for many kinds of products, including those with heterogeneous characteristics (Keraita and Kim, 2007). Combinational weighers have a number of weighing hoppers that statically weigh the product; these weight data are fed into a computer, which calculates all of the possible combinations of product weights in order to dispense the best combination (closest match to target weight) to a packaging machine. The weighing system consists of three elements, namely: A system to automate the product feed to the weighing stations (depending on the layout of the machine, the feed system is configured either in a radial or in line construction); a system to collect the product and feed it into a weighing hopper (this system consists of a set of hoppers, commonly known as feed hoppers); and a set of weighing hoppers.

1.2 Problem description

A multihead packing process performs an operation by choosing a subset H' from set H of the current n hoppers to produce a food package. The basic model of the automated packing system consists of n weighing hoppers. A quantity of food is placed in each hopper i, $(i = 1,2, \dots, n)$ (Karuno et al., 2007), and the weight signal is transmitted to the built-in computer. The computer calculates the combinations of weights that come closest to the desired weight T, and the combination of the closest weights is ejected from the corresponding hoppers. The resulting empty hoppers are supplied with new quantities of food. The computer continuously repeats this process until it obtains the number of packages (Q) needed, one by one. Based on our experience in this field, multihead weighers are able to produce between 50 and 250 packages per minute, depending on the specific setting.

The number of possible different hopper subsets H' depends on the number k of hoppers to be combined each time. In fact, as Imahori et al. (2011) pointed out, the optimization problem that focuses on minimizing the difference between the actual and target package weight is equivalent to the NP-complete subset-sum combinatorial problem (Garey and Johnson, 1979) when k is neither previously fixed nor constant.

This paper deals with the case where the number of hoppers k to be combined in each packing operation is constant and fixed in advance. In addition, it assumes that the weights in the hoppers follow a normal probability distribution.

An additional point to take into account in this kind of packaging process is that a given quantity of product can remain for a long time in its corresponding hopper until it is chosen for packing. This can be a problem when handling a product that will deteriorate quickly, such as, a frozen product. One possible way to tackle this problem is to monitor and control the load time spent in each hopper, which can be done by assigning a priority coefficient P_i to each hopper, as suggested by Karuno et al. (2007). The priority P_i measures the duration of the load in hopper i before it is chosen for packing, and it can be calculated as follows: Let ℓ denote the current iteration number of the packing operation, and let ℓ_i denote the iteration number at which weight i was thrown into the i-th hopper when the hopper was empty. Therefore, $P_i = \ell - \ell_i + 1$ expresses the residence time (in number of packing operations) of weight i in its hopper. Note that $1 \leq \ell \leq \mathrm{Q}$. The idea is that hoppers with higher priorities at a given moment should be more likely to be chosen for emptying in that packing operation.

In real-world engineering, optimization problems are often characterized by the presence of multiple objective functions. Multi-objective optimization involves the simultaneous optimization of two or more conflicting objectives. A considerable amount of research has been conducted in this area in the past thirty years. The principles, implementation and applications of multi-objective optimization models in engineering optimization problems can be followed in Jaimes and Coello (2008), Marler (2009), Seng and Rangaiah (2009), Rangaiah and Bonilla-Petriciolet (2013), Collette and Siarry (2013), Liu and Papageorgiou (2013), and Zavala et. al (2014).

1.3 Previous related works

The scientific references in the field of multihead weighing are scarce. Basically, conference papers, patents, and commercial documentation of manufacturers of this type of machine are available. With these restrictions, a state-of-the-art for improving the multihead weighing technology is presented. Some authors have studied the possibility of improving multiweighing procedures in packing processes' performance. For example, Salicrú et al. (1996) and Barreiro et al. (1998) proposed the use of the percentage variability reduction index for the reduction and control of production process variability. Keraita and Kim (2006) investigated the optimum scheme for the determination of the operation time of line feeders in automatic combination weighers. Keraita and Kim (2007) proposed a weighing algorithm for multihead weighers based on bit operation. In Karuno et al. (2007), a second objective called "priority" is introduced. They formulated the problem as a bi-criteria optimization problem, and proposed an algorithm based on dynamic programming. The proposed approach aimed to minimize the maximum duration in the system of items heuristically (maximum priority), while making the total weight of each package as close to the target weight as possible. Some authors (Imahori et al., 2011; Imahori et al., 2012; Karuno et al., 2013; Karuno and Tateishi, 2014) have studied the possibility of improving the bi-criteria optimization model proposed by Karuno et al. (2007). Other authors, such as Imahori et al. (2012) and Karuno et al. (2010), investigated different types of actual packing operations. In these investigations, several algorithms are developed for double-layered food packing systems and duplex packing systems, where in an operation two disjointed subsets are simultaneously selected to produce two packages.

1.4. Objectives and structure

This paper focuses on improving the quality of a multihead packing process in a two-fold way: the quality of the process and the sensorial quality of the product. Specifically, we propose a bi-objective programming approach in order to simultaneously deal with both the criterion of minimizing the difference between the target and actual package weight in absolute value (improving the quality of the process), and the criterion of maximizing the total priority of the chosen combination of hoppers $\sum_{i\in H'} P_i$ (improving the sensorial quality). More precisely, we use compromise programming (Yu, 1973; Zeleny, 1973; Marler, 2009; Collette and Siarry, 2013) as a tool to determine the combination of hoppers that comes as close to optimizing both criteria at the same time as possible in each iteration or packing operation, that is, each time a new package has to be made. An enumerative — and, therefore, exact— procedure is proposed to determine the best hopper combination according to this approach. Our procedure includes an easy-to-implement way to dynamically adjust the relative importance of each objective (weight and priority). Further information about multi-objective programming and other multicriteria techniques can be found in Marler and Arora (2004), Ehrgott (2005), Branke, J. et al. (2008) and Rangaiah and Bonilla-Petriciolet (2013). As mentioned above, Karuno et al. (2007) introduced the use of multicriteria techniques in automated packing systems. Later in this paper we will point out the differences between their approach and ours.

This paper is organized as follows: In section 2, the problem formulation is presented. In section 3, the bi-objective algorithm is explained. Section 4 shows the computational experiments. In section 5, we offer the conclusions of this work.

2 Problem Formulation

2.1 Notation

Problem parameters and symbols:

- H: Set of the current n hoppers.
- H': Subset of the current k hoppers.
- ℓ: Current iteration number of the packing operation.
- ℓ_i: Iteration number at which weight i was thrown into the i-th hopper when the hopper was empty.
- P^{MAX}: Maximum allowed priority for any hopper.
- k: Number of hoppers to be combined in each packing operation.
- X_i : Real weight in hopper $i \in \{1, \dots, n\}$. We assume that each weight follows a normal probability distribution $\mathrm{N}(\mu_j, \sigma)$, with $j \in \{1, \dots, 5\}$, depending the subgroup to which the hopper belongs (see below).
- μ_j : Average weight for hopper subgroup $j \in \{1, \dots, 5\}$.
- σ: Standard deviation of weights in every hopper. It is equal for all the n hoppers.
- CV: Percentage value used to calculate the standard deviation of weights in every hopper (σ).

- *Delta*: Real value used to establish μ_j for hopper subgroup $j \in \{1, \dots ,5\}$.
- n_j: Number of hoppers in hopper subgroup $j \in \{1, \dots ,5\}$.
- d: Qualitative factor. It determines how the total number of hoppers (n) are distributed in each hopper subgroup $j \in \{1, \dots ,5\}$.
- $Z_{\alpha/2}$: Critical value of the standard normal probability distribution $N(0,1)$ for a significance level α.
- T: Target weight for a single package, which is assumed to be a positive number.
- W: Total weight. It is calculated as the sum of the weights provided by the selected combination of k hoppers.
- z_1: Difference (in absolute value) between the target weight (T) and the actual weight of the selected combination (W).
- z_2: Sum of the priorities P_i into k hoppers.
- Q: Total number of packages needed.

Algorithm parameters and symbols:

- θ: Relative weight or importance of the priority objective. It is dynamically adjusted in each iteration.
- P_i: Positive integer priority in hopper $i \in \{1, \dots n\}$.
- $z_1^{\min}$: Minimum difference (in absolute value) between the target weight (T) and the actual weight of the selected combination (W).
- $z_1^{\max}$: Maximum difference (in absolute value) between the target weight (T) and the actual weight of the selected combination (W).
- $z_2^{\min}$: Minimum sum of the priorities P_i in k hoppers.
- $z_2^{\max}$: Maximum sum of the priorities P_i in k hoppers.
- S: The set of all valid combinations in each package operation.

2.2 Sources of variability

The total weight of packages W produced by a multihead weighing process can be seen as a random variable. The variability depends on the value of several process and operational parameters.

As mentioned previously, the package weight is the result of the sum of the weights contained in the subset of k hoppers selected to form the package, assuming that the weights in each hopper are normally distributed. Particularly, if all the hoppers are independently filled according to the same distribution $N(\mu, \sigma)$ and the k hoppers are randomly selected in each packing operation, then the weight of the packages would follow a normal distribution $N\left(k\mu, \sqrt{k}\sigma\right)$, where the average package mean weight $k\mu$ is expected to equal the target T. The value of $\sqrt{k}\sigma$ (the standard deviation if hoppers are selected at random) is considered to be an index of quality in the packaging process. However, the subset of hoppers to be discharged H' is actually not selected at random, but rather in a driven way, so that the total weight $W = \sum_{i \in H'} X_i$ is as close to T as possible. Therefore, $\sigma_{\text{package}}^2 = \text{VAR}\left(\sum_{i \in H'} X_i\right)$. This proposal was also presented by Salicrú et al. (1996) and Barreiro et al. (1998).

Note that the parameter k is one of the factors that can clearly affect the final variability of the product, as it limits the number of possible hopper combinations (i.e., $\binom{n}{k}$, the number of possible different subsets of size k from a set of n hoppers) in each packing operation. For the same reason, the number of total available hoppers n is also a source of variability in the package weight. Obviously, the existing variability in each hopper (which can be represented by the standard deviation (σ) will somehow affect the final package variability. It can actually be expressed in a dimensionless way by means of the coefficient of variation CV (see later in subsection 4.1).

Furthermore, let us consider the general case where each hopper i is expected to be filled with a different average quantity of food μ_i (instead of a common value μ). In this case, the degree of variability between these average hopper weights $\mu_1, \dots, \mu_n$ is expected to somehow be related to the final package variability. In this paper we will explore the case where several hoppers weights are set in such a way that they share the same value for μ_i (see subsections 3.1 and 4.1 later), as this has been shown to be an efficient strategy to reduce package variability (Barreiro et al., 1998; Keraita and Kim, 2007; Pulido-Rojano and García-Díaz, 2016). More precisely, Pulido-Rojano and García-Díaz (2016) showed that the strategy of dividing hoppers into 5 groups with different average filling weights and a certain deviation among these average weights was a better strategy than an equal supply of the product to all hoppers. Therefore, in this paper we will use a filling strategy divided into "5 subgroups" of hoppers.

These and other parameters, such as the maximum allowed priority for any hopper (which will be denoted as P^{MAX}) should be considered as sources of variability in a multihead packaging process. Later, in Section 4, we perform numerical experiences to assess the real influence of these factors on the final package quality, in terms of variability from the target T, when using the hopper selection algorithm we present in subsection 2.5.

2.3 Problem Constraints

The decision problem we address in this paper is how to choose *the best* (or *the most appropriate*) combination of k hoppers in each packing operation. In order to make the problem more realistic, we state two additional constraints that any k-hopper combination H' should meet in order to be eligible:

1. $\left| T - \sum_{i \in H'} X_i \right| \leq \mathrm{Z}_{\alpha/2}\sqrt{k}\sigma$, which avoids any k-hopper combination that would produce a package too far from the target T. $\mathrm{Z}_{\alpha/2}$ represents the critical value of the standard normal probability distribution $N(0,1)$ for a significance level α. This is called the *confidence level* constraint.
2. $P_i \leq P^{\text{MAX}}, \ \forall\, i \in H'$, which means that the selected k-hopper combination must not involve any hopper containing food that exceeds the maximum allowed residence time (in terms of priority, as defined in section 1.2).

Every k-hopper combination that simultaneously meets conditions 1 and 2 is said to be a *valid* combination. The *set* of all *valid combinations* in each package operation will be denoted by S, $S = \{ H' \subseteq H \ | \ |H'| = k$ and H' meets conditions 1 and 2 $\}$.

Thus, the subset H' of hoppers selected to form the package has to belong to S. In the following subsections, we describe our proposal about how to make this decision.

2.4 *Objective functions*

In line with what was introduced in section 1.2, the selection of hoppers H' to be discharged in each package operation should address the following two objectives in order to be considered a *good* combination:

- First objective: To try to make the difference between the real package weight $W = \sum_{i\in H'} X_i$ and the target weight T as small as possible. This will be expressed through the following objective function which will be minimized: $z_1 = |\ T - \sum_{i\in H'} X_i\ |$.
- Second objective: To make those hoppers that have not been discharged for a long time (i.e., with a long residence time) more likely to be selected. In terms of priority, this can be achieved by maximizing the following function: $z_2 = \sum_{i\in H'} P_i$, which represents the aggregated priority of a given k-hopper combination. In the next subsection we develop our proposal about how to simultaneously take into account these two objectives in each packing operation.

2.5 *Bi-objective approach*

We propose using a single weighted performance or utility function that combines information about the two objectives or criteria being considered in this work (weight and priority), where the relative weight or importance of each objective is dynamically adjusted in each iteration or packing operation. More precisely, in each iteration, our approach consists of looking for the k-hopper combination that minimizes a sort of "distance" to the so-called *utopia* or *ideal point* $\left(\mathrm{z}_1^{\min}, z_2^{\max}\right)$ in the criterion space, where $z_1^{\min}$ is the minimum possible difference (in absolute value) between the target and the actual weight of a k-hopper combination for the current hopper loads, and $z_2^{\max}$ is the maximum possible aggregated or total priority, that is:

$$z_1^{\min} = \min_{H' \in S} |\ T - \sum_{i\in H'} X_i\ |\ , \tag{1}$$

$$\text{and} \quad z_2^{\max} = \max_{H' \in S} \sum_{i\in H'} P_i. \tag{2}$$

Therefore, $z_1^{\min}$ and $z_2^{\max}$ are the respective optimal values for the two objectives being considered for the current hopper setting, if each of them was optimized separately. Prior to calculating the Euclidean distance (D) from a given solution to the ideal point, each of these two values (difference from target weight and priority) is normalized and then assigned a relative weight of $(1-\theta)$ and θ, respectively, so that the final aspect of the function whose value is intended to be minimized is the following:

$$D = \sqrt{(1-\theta)\left(\frac{z_1 - z_1^{\min}}{z_1^{\max} - z_1^{\min}}\right)^2 + \theta\left(\frac{z_2 - z_2^{\max}}{z_2^{\max} - z_2^{\min}}\right)^2}\ , \tag{3}$$

where $z_1^{\max}$ and $z_2^{\min}$ are respectively defined as the maximum difference from the target weight and the minimum total priority in the current set of valid k-hopper combinations.

The parameter θ is updated in each iteration. The idea is that the objective of selecting a k-hopper combination with a high aggregated priority becomes more important as the maximum current hopper priority gets close to the maximum allowed priority P^{MAX}. With this in mind, θ can be defined as:

$$\theta = \frac{1}{P^{\text{MAX}} - \max_{i \in H} P_i + 1} \ . \qquad (4)$$

Therefore, during the first iterations, in which all the hoppers are expected to have low priority values, the value of θ will remain relatively small, and so, the objective of minimizing the difference from the target packet weight will be assigned a higher importance. As packages production progresses, the closer the maximum hopper priority gets to P^{MAX}, the larger θ will become and the greater the importance given to the priority objective. The combination of hoppers that minimizes the distance to the ideal point is known to be an efficient or nondominated solution (Marler and Arora, 2004), which means that there is no other valid combination of k hoppers that is at least as good with regard to (at least) one of the objectives (weight or priority) and strictly better in the case of the other objective (Ehrgott, 2005). This is generally regarded as a basic desirable property for the solution(s) returned by multi-objective techniques.

This bi-objective approach makes it possible to easily select a valid k-hopper combination that is reasonably close to optimizing both objectives being considered. In order to show this, a simple practical example and further complete numerical experiences are presented in sections 3 and 4, respectively.

Our approach goes one step further than the one by Karuno et al. (2007), who only show how to generate different nondominated solutions, without specifying which of them should be selected in each packaging operation, whereas we are suggesting a way to automatically determine a compromise solution within the nondominated ones. Moreover, although Karuno et al. (2007) are not following a purely lexicographic approach (i.e., a subordinated objective importance structure), they are still considering the weight objective to be more important than the priority objective (namely, they look for solutions that reduce residence time without moving too far away from the target weight, rather than the other way around). Conversely, we consider both objectives to potentially be equally important, as explained above. In addition, the respective contexts for the two studies are quite different. To be precise, Karuno et al. (2007) look for combinations of any number of hoppers in each iteration; that is, the parameter k is not fixed in advance and can change in each iteration; also, they consider the values for the weights X_i to be integers and uniformly distributed, whereas this paper deals with real-values following a normal distribution.

3 Bi-objective Algorithm

In this section, our bi-objective algorithm is outlined, along with a numerical example showing a sample iteration. This algorithm can be implemented in the software systems installed in the control unit of a multihead weigher.

As suggested by previous studies (see subsection 2.2) aimed at a better performance, the set of hoppers will be divided into five subgroups, with the hoppers in each of them being filled with a different average amount of product. More precisely, each hopper in the third subgroup will be assigned an average weight of $\mu_3 = T/k$, whereas the rest of the hopper subgroups will be assigned the same average weight plus or minus a certain shift, according to a given pattern. The mean shift in groups 1, 2, 4 and 5 is determined by means of two parameters, $Delta$ and min_Delta, to be provided to the algorithm (see Step 1). The rest of the details about the algorithm have been introduced in sections 1 and 2.

3.1 Step-by-step algorithm

- **Input:**
  - n: Total number of hoppers. $n > 0$.
  - k: Number of hoppers to be combined in each packing operation. $2 \leq k < n$.
  - T: Target weight. $T > 0$.
  - $n_1, \ldots, n_5$: Number of hoppers in each subgroup of hoppers. $n_j \geq 0$, $\forall j = 1, \ldots, 5$; $\sum_{j=1}^{5} n_j = n$.
  - σ: Standard deviation of the weight to be provided to each hopper. $\sigma > 0$.
  - $Delta$: Relative mean shift for hoppers in subgroups 1 and 5 with regard to subgroup 3. $Delta > 0$.
  - min_Delta: A value such that $Delta - min\_Delta$ is the relative mean shift for hoppers in subgroups 2 and 4 with regard to subgroup 3. $0 < min\_Delta \leq Delta$.
  - P^{MAX}: Maximum allowed priority (number of iterations without being chosen) for any hopper. $P^{\text{MAX}} \geq 1$.
  - Q: Total number of packages to be produced. $Q \geq 1$.
- **Step 1. Initialization.**
  - Assign each hopper to a subgroup, so that the number of hoppers in subgroup j is n_j, for all j.
  - *Calculate the average hopper weight to be provided to each hopper subgroup.* $\mu_1 = T/k - Delta \cdot \sigma$; $\mu_2 = T/k - (Delta - min\_Delta) \cdot \sigma$; $\mu_3 = T/k$; $\mu_4 = T/k + (Delta - min\_Delta) \cdot \sigma$; $\mu_5 = T/k + Delta \cdot \sigma$.
  - *Set initial values for weights and priorities for each hopper.* $X_i = 0$, $P_i = 0$, $\forall i = 1, \ldots, n$.
  - *Set an initial value for the number of produced packages so far.* $q = 0$.
- **Step 2. New packaging operation.** Initialize $z_1^{\min} = +\infty$, $z_1^{\max} = -\infty$, $z_2^{\min} = +\infty$, $z_2^{\max} = -\infty$, $D_{\min} = +\infty$, $H'_{\min} = \emptyset$.
- **Step 3.** *Refill all empty hoppers and update priorities.* For all hopper i in subgroup j such that $X_i = 0$: Let $X =$ random value from a $N(\mu_j, \sigma)$ distribution. For all hopper i: Let $P_i = P_i + 1$.
- **Step 4.** *Discard and discharge (out of the package) any hopper that does not meet the priority constraint.* For all hopper i such that $P_i > P^{\text{MAX}}$: Let $X_i = 0$, $P_i = 0$.

- **Step 5.** *First evaluation of* all *valid combinations, in order to calculate* $z_1^{\min}$ *,* $z_1^{\max}$ *,* $z_2^{\min}$ *,* $z_2^{\max}$. For all k-hopper combination H' such that it does not contain any hopper i with $P_i = 0$ and such that $\left| T - \sum_{i \in H'} X_i \right| \leq Z_{\alpha/2}\sqrt{k}\sigma$:
  - Calculate z_1 and z_2.
  - If $z_1 < z_1^{\min}$, then $z_1^{\min} = z_1$.
  - If $z_1 > z_1^{\max}$, then $z_1^{\max} = z_1$.
  - If $z_2 < z_2^{\min}$, then $z_2^{\min} = z_2$.
  - If $z_2 > z_2^{\max}$, then $z_2^{\max} = z_2$.
- **Step 6.** *Check that the set of valid combinations is not empty.* If $z_1^{\min} = +\infty$ then: *(there is no valid combination; all hoppers must be discharged and refilled)* For all hopper i: Let $X_i = 0,\ P_i = 0$; go to Step 2. Otherwise: go to Step 7.
- **Step 7.** Calculate θ. *(relative importance of the priority objective; it is recalculated before each packing operation).*
- **Step 8.** *Second evaluation of* all *valid combinations, in order to select the one that minimises the performance function* D. For all k-hopper combination H' such that it does not contain any hopper i with $P_i = 0$ and such that $\left| T - \sum_{i \in H'} X_i \right| \leq Z_{\alpha/2}\sqrt{k}\sigma$:
  - Retrieve z_1 and z_2 for H'. Calculate D. If $D < D_{\min}$ then let $D_{\min} = D,\ H'_{\min} = H'$.
- **Step 9.** *The k-hopper combination minimising D is the one that has to be selected to produce the package.* **Return** $H'_{\min}$ *(as the set of hoppers to be combined to create the $(q+1)$-th package).* For all hopper i belonging to $H'_{\min}$: *(it is discharged into the package)* Let $X_i = 0,\ P_i = 0$.
- **Step 10.** *Update the number of packages produced and check whether the process is finished or not.* Let $q = q + 1$. If $q < Q$ then go to Step 2; otherwise, END.

Steps 5 and 8 of this algorithm reveal the enumerative nature of our proposal. Each of the two steps can easily be implemented by using nested loops (as many loops as k). More precisely, because every feasible solution (i.e., valid k-hopper combination) is evaluated in each iteration of the algorithm, our approach can be said to follow an explicit enumerative strategy (or exhaustive search), as announced in the introduction section. In particular, the number of combinations to be evaluated in a single iteration equals $\binom{n}{k} = n!/(k!\,(n-k)!)$ at most (because some hoppers can be discarded in Step 4 due to the priority constraint). Although it is a simple strategy, this allows our bi-objective algorithm to be considered an exact (not heuristic) search (Michalewicz and Fogel, 2004). This establishes another difference between our bi-objective approach and the one by Karuno et al. (2007), who propose a heuristic strategy (in the sense that they do not necessarily generate all the nondominated solutions to the problem, but only those that are close enough to the target weight). The computational cost of generating and evaluating all the valid hopper combinations can be accepted by our algorithm because it only considers combinations of k hoppers, with the parameter k being fixed in advance.

Notice that step 6 of the algorithm describes a situation in which all hoppers should be discharged in order to avoid producing packages that would not meet the quality requirements for the final product in terms of weight. In practical terms, all of this discharged product could be taken and reused in the process again, for instance. In any case, this *full* discharge happens very infrequently (as can be seen later in Section 4.4) and, therefore, would not significantly affect the final cost of the packaging process.

3.2 Numerical example

For clarity, we show an example of how the iterations of the proposed algorithm work. Assume T = 500 grams and σ = 12.50 grams, and suppose that we are choosing k = 4 out of n = 16 hoppers in each iteration, which are distributed as stated in step 1 of the algorithm, with $n_1 = 3, n_2 = 3, n_3 = 4, n_4 = 3$ and $n_5 = 3$. Let us also suppose that the value of exchange is $Delta$ = 1.5, min_Delta = 0.5 and the maximum allowed priority for any hopper is P^{MAX} = 10. In these conditions the filling setting would be: μ_1 = 106.25 grams, μ_2 = 112.50 grams, μ_3 = 125 grams, μ_4 = 137.50 grams and μ_5 = 143.75 grams. Figure 1 shows the situation at iteration 1000 of this specific example.

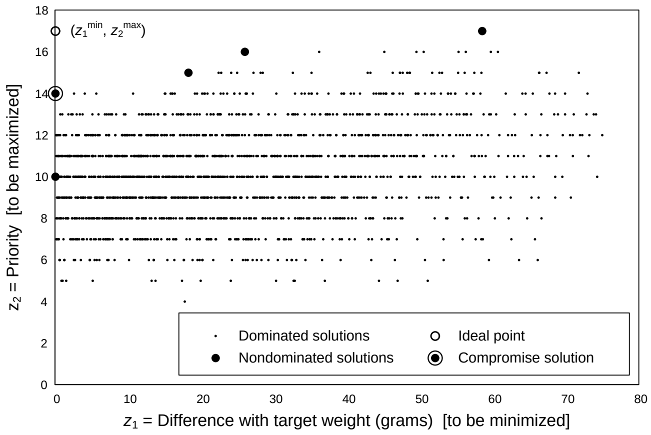
Combining different sets of 4 hoppers results in a total of 1790 valid combinations, after discarding those that do not meet the condition stated in steps 4 and 6. In the figure, each of these solutions is represented according to its coordinates in the decision or criterion space, with the horizontal and vertical axes corresponding to the difference from the target weight and the total priority, respectively.

The lowest difference with regard to the target weight, $z_1^{\min}$ = 0.05481 grams, is achieved by combining hoppers 1, 2, 10 and 14, whereas the highest aggregated priority $z_2^{\max}$ = 17 corresponds to hoppers 6, 10, 11 and 13. As a direct consequence, the ideal point is ($z_1^{\min}$ = 0.05481, $z_2^{\max}$ = 17). As shown in the chart, there are five nondominated or efficient solutions, which form the so-called Pareto set or Pareto frontier, and the two aforementioned solutions are part of it. The maximum current hopper priority in this iteration is 5, and therefore $\theta = 1/(10 - 5 + 1) = 0.1667$, according to (4). This means that, in this very specific iteration, the ratio of importance of the two objectives (weight and priority) is $(1 - \theta): \theta = 5: 1$ (namely, the objective of getting a package close to the target weight is five times more important than the objective of selecting a 4-hopper combination with a high total priority).

Each of the 1790 solutions is compared to the performance function D (weighted normalized distance to the ideal point), as defined in (3). Notice that it would only be necessary to check the value of D in the five nondominated solutions, as the point that minimizes the value of D always belongs to the Pareto set in compromise programming. However, numerically finding out whether each of the 1790 valid combinations is dominated or not would be equivalent to calculating D in each of them, in terms of computational effort.

In this example, the minimum value of D is given by choosing hoppers 1, 5, 6 and 11, which corresponds to a difference from the target weight of only 0.06173 grams (the second best combination, with regards to this objective) and a total priority of 14 (to be more precise, one of the priorities of these four hoppers that are going to be discharged is equal to 5, which was the maximum priority in this iteration). This example illustrates how compromise programming succeeds in balancing the objectives being considered in an automatic and reasonable way.

Figure 1 Feasible solutions set and Pareto frontier (nondominated solutions) for the numerical example in Section 3.2, showing a snapshot of a specific iteration of the proposed bi-objective algorithm. Each point represents a valid hopper combination. The coordinates for each point represent the difference of that combination with regards to the target weight (z_1) and the total priority of that combination (z_2). The five ‘big’ black dots represent the nondominated solutions (candidates to be returned as the selected hopper combination) and the ‘empty’ dot represents the ideal solution (a theoretical one combining the best of z_1 and z_2). The hopper combination that is finally selected (i.e., the compromise solution) is the one that is closer to the ideal point, using the distance *D* defined in Section 2.5.



4 Computational Experiments

In this section, an extensive set of numerical simulation experiments is presented to study the performance of the proposed bi-objective algorithm.

4.1 Tested process and operational parameters

In order to find the optimum operative conditions to minimize the variability in the total weight (W), a design of experiments (DOE) was carried out. The response variable used is the "coefficient of variation of the package" ($\mathrm{CV}_{\mathrm{package}} = \frac{\sigma_{\mathrm{package}}}{\mu_{\mathrm{package}}} \cdot 100\%$). This DOE takes into consideration a large number of possible productive configurations of the multihead weigher. The factors and their levels to study in the DOE are shown in Table 1 and 2. In this case, the design of experiments is a balanced factorial design of fixed effects factors.

As has been seen, the factor “Value of exchange” (*Delta*) helps to adjust the filling setting during the packaging operation. This means that *Delta* makes it possible to observe the influence of fixing the average weights for the subgroups of hoppers (μ_j) in many different cases. In this way, *Delta* (As has been seen in step 1 of the algorithm) involves voluntary changes in the supply of products to different subgroups of hoppers, except for n_3.

On the other hand, the factor d represents the distribution of the hoppers in each subgroup and its levels define the number of hoppers that will be in each. Therefore, an *Equal* level means that the number of hoppers in the subgroups is as homogeneous as possible. In the *Central* level, the largest number of hoppers is concentrated in the subgroup of hoppers whose filling objective has not been modified (n_3). In the *Extreme* level, the largest number of hoppers is concentrated in the subgroups of hoppers where the filling objective has been modified (n_1, n_2, n_4 y n_5).

The coefficient of variation (CV) is used to calculate the standard deviation of the weights in every hopper (σ) as an input in the packaging process, e.g., if $\mathrm{CV} = \frac{\sqrt{k}\sigma}{T} \cdot 100\% = 5\%$, T = 500 and k = 4. Theoretically, we have $\sqrt{k}\sigma$ = 25 and, therefore, σ = 12.50. The above allows the simulation of different scenarios of the standard deviation of weights in every hopper. However, it does not mean that $\sqrt{k}\sigma$ will be the actual variability obtained in the package produced through our proposed approach. Thus, calculating all the combinations of factors will result in 37800 treatments, which were simulated 10000 times each, i.e., 10000 units of packaged products for each treatment. The conclusions of the analysis are presented in subsection 4.3.

Table 1 Levels of the factors studied

Factor	**Levels**
Number total of hoppers (n)	8,10,12,14,16
Number of hoppers to be combined (k)	2,3,4,5,6,7
Target weight (T)	125, 250,500, 1000, 2000 g.
Coefficients of variation (CV)	1%, 2.5%, 5%
Distribution of weighing hoppers (d)*	Equal, Center, Extreme
Value of exchange ($Delta$)	0.0,0.5,1.0,1.5,2.0,2.5,3.0
Maximum allowed priority for any hopper (P^{MAX})	10, 30, 50, 100

* See Table 2

Table 2 Distribution of weighing hoppers for each subgroup

		Equal					**Center**					**Extreme**				
		n_1	n_2	n_3	n_4	n_5	n_1	n_2	n_3	n_4	n_5	n_1	n_2	n_3	n_4	n_5
Total number of hoppers (n)	8	1	2	2	2	1	1	1	4	1	1	3	1	0	1	3
	10	2	2	2	2	2	1	1	6	1	1	4	1	0	1	4
	12	3	2	2	2	3	1	1	8	1	1	4	2	0	2	4
	14	3	3	2	3	3	1	1	10	1	1	5	2	0	2	5
	16	3	3	4	3	3	1	1	12	1	1	6	2	0	2	6

4.2 Performance measures

The most important calculated parameters, as a measure of performance to evaluate our bi-objective approach, are: Average weight of the total number of packages produced ($\mu_{package}$), the standard deviation of the total number of packages produced ($\sigma_{package}$), percentage of discharge due to the confidence level (DCL), the number of hoppers discarded by priority for each iteration (HDP), the average maximum priority for each hopper (AMP), and the *Trade-off* values.

4.3 Statistical analysis

Analysis of variance (ANOVA) was used to determine the statistical significance of the factors and their interaction. The ANOVA procedure assumes that the observations are normally and independently distributed, with the same variance for each treatment or factor level. In order to verify the statistical validity of the results and ascertain what the best configuration is, we performed a multifactor ANOVA where the response variable is $\log(CV_{package})$. The transformation of the response variable was necessary in order to ensure compliance with the ANOVA's three important hypotheses; normality, homogeneity of variance and independence of the residuals (Montgomery, 2012).

The ANOVA confirms that all the main effects and interactions of the factors are statistically significant when $\mathrm{CV}_{package}$ is measured, except for T. The CV factor is the most significant and it does not interact with any of the other factors. Moreover, significant interactions between *Delta* and *k, Delta* and *d*, k and n, and d and k are highlighted (the ANOVA table is not shown due to lack of space).

The mean plots and least significant differences (LSD) intervals for k, *Delta* and d are shown in Figure 2. As Figure 2 shows, some deductions can be made. When the number of weighing hoppers combined reaches values of k = 4 or k = 5, low $\mathrm{CV}_{package}$ values are obtained. A value of exchange (*Delta*) of 2.0 reduces the $\mathrm{CV}_{package}$ value. A homogeneous distribution of hoppers (*Equal*) in subgroups causes a decrease in the $\mathrm{CV}_{package}$.

Figure 3 shows the interaction plots for significant factors. The analysis of interaction plots shows some interesting results. The value of exchange (*Delta*), which results in a reduction in the $\mathrm{CV}_{package}$, depends on the way the weighing hoppers are distributed and the number of weighing hoppers combined (k). The values of exchange (*Delta*) of 1.5, 2.0 and 2.5 provide the best results when the total number of weighing hoppers (n) is the highest.The values of exchange (*Delta*) of 1.5 or 2.0 are statistically equivalent and provide a decrease in $\mathrm{CV}_{package}$ when the maximum allowed priority for any hopper (P^{MAX}) reaches its highest value. Note that for k = 2, a *Central* distribution of weighing hoppers is preferred. A homogeneous distribution of hoppers (*Equal*) is able to reduce the $\mathrm{CV}_{package}$ when multihead machines with a high number of weighing hoppers (n) are employed and the maximum allowed priority for any hopper (P^{MAX}) takes the value of 100. A number of weighing hoppers combined (k) of 6 or 7 is statistically equivalent and provides the best results in reducing the $\mathrm{CV}_{package}$ when the total number of weighing hoppers (n) is used. Note that the best value of k also depends on the value n, which is confirmed by the strong interaction between these factors.

Figure 2 Mean plots and LSD intervals for significant factors as a function of $\log(\mathrm{CV}_{paquete})$.

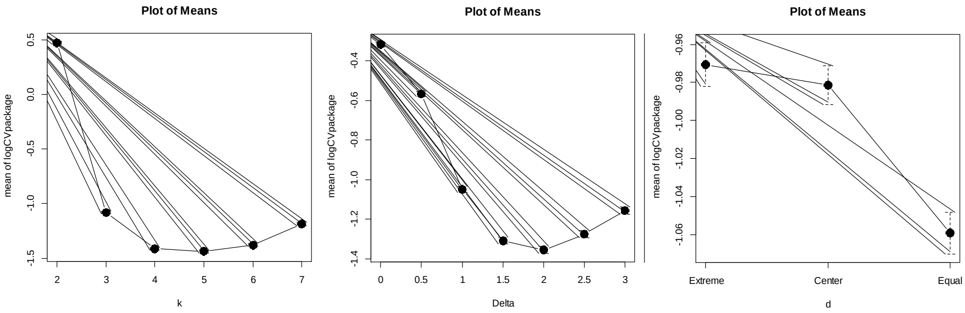
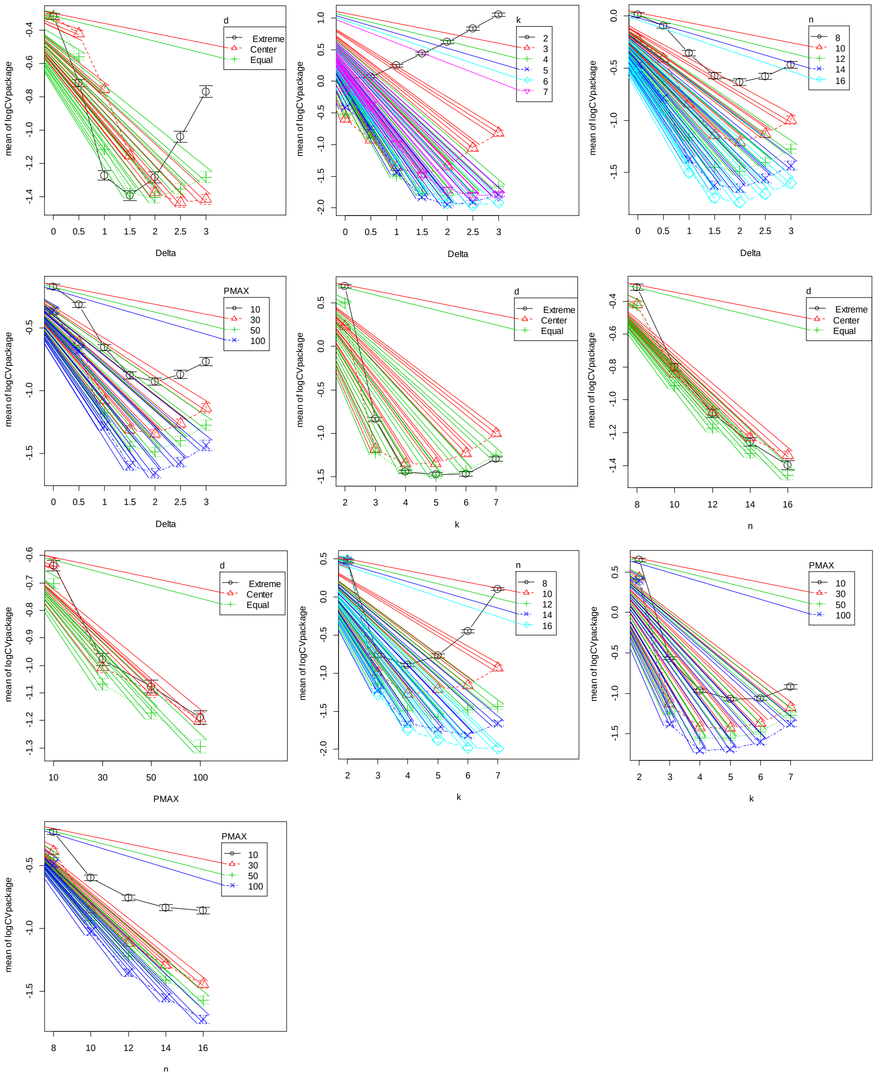


Figure 3 Interactions plots and LSD intervals for significant factors as a function of $\log(\mathrm{CV}_{paquete})$.



Based on this analysis, we can now obtain optimum operating conditions that minimize the response variable. The final levels for the factors are shown in Table 3.

Table 3 The best operational conditions for the multihead weighing process

Factor	**Best Level**
Number total of hoppers (n)	16
Number of hoppers to be combined (k)	4
Coefficients of variation (CV)	1%
Distribution of weighing hoppers (d)	Equal
Value of exchange ($Delta$)	2.0
Maximum allowed priority for any hopper (P^{MAX})	100

4.4 Numerical results and discussion

As a supplement to the statistical analysis described in 4.3, a summary of results is presented that makes it possible to analyse the behaviour of performance parameters during the packaging process when the proposed bi-objective algorithm is run. These results were compared to the outputs in a mono-objective approach. Taking into account that the levels of certain factors, due to the high interaction between them, depend on the levels of other factors, we decided to summarize the results by using the following inputs.

Consider a packaging process under control where we aim to obtain a target weight of 500 grams. The computational experiment was performed using the following case: n = 16, $Delta$ = 2.0, min_Delta = 0.5, P^{MAX} :{10, 30, 50, 100} and k:{2, 3, 4, 5, 6, 7} hoppers. An *Equal* distribution of the weighing hoppers in each subgroup was used, so: $n_1 = 3$, $n_2 = 3$, $n_3 = 4$, $n_4 = 3$ and $n_5 = 3$. To calculate the standard deviation of weights in every hopper (σ) as an input in the packaging process, CV:{1%, 2.5%, 5%} were used.

A way of measuring the loss in the increases in variability in the package in exchange for a decrease in the duration of the product in the hopper is to calculate the *Trade-off* value, as follows:

$$\textit{Trade-off} = \left| \frac{\Delta\ \sigma_{\text{package}}}{\Delta \text{ Average maximum priority /Hopper}} \right|, \tag{5}$$

where "$\Delta\ \sigma_{\text{package}}$" is a measure of the shift in the package variability, and "Δ Average maximum priority /Hopper" is a measure of the change in the average maximum priority for each hopper for a mono-objective approach and our bi-objective approach.

Tables 4 and 5 show the performance parameters for the bi-objective approach proposed and the mono-objective approach. P^{MAX} values allow the monitoring of the evolution of the performance parameters of the process. P^{MAX} represents the maximum residence time of the product in a hopper. For instance, for a multihead weigher with a capacity of fifty packages per minute and P^{MAX} = 100, the residence time is calculated as follows: 50 packages/60 seconds is equivalent to 1.2 seconds/package. Therefore, 1.2 seconds/package·100 packages = 120 seconds.

In table 4, the results show that when the priority P_i is considered, the variability of the package increases and the duration of the product in the hopper decreases (as expected). It can be observed that in the bi-objective approach, there were no full discharges for the confidence level of any of the P^{MAX} values considered. The above confirms that, regardless of the increased variability in the weight of the packaged product due to low values of maximum allowed priority (P^{MAX}), at least one of the weights obtained from all the combinations in each iteration was within the confidence level of 99.73%. Note that discharges of hoppers due to exceeding the maximum allowed priority only occurs when P^{MAX} = 10 and k = 2. The largest decrease in the duration of the product in hopper and the largest increase in variability are produced when we use k = 2 for any of the P^{MAX} values. In fact, in these cases the variability of the packaged product is higher than the variability when the packaging process is performed randomly. Note that in each case analysed, when k = 7, the lowest values for the $CV_{package}$ are obtained in both approaches (bi-objective and mono-objective). This shows the strong interaction between the number of weights combined (k) and the total number of hoppers (n).

Table 4 Simulation results from the bi-objective algorithm for different values of the maximum allowed priority (P^{MAX}) compared to results from the mono-objective approach. See section 4.2 for a further explanation of the performance parameters listed.

		Bi-objective approach																				**Mono-objective approach**		
		$\boldsymbol{P^{\mathrm{MAX}}}$ **= 10**					$\boldsymbol{P^{\mathrm{MAX}}}$ **= 30**					$\boldsymbol{P^{\mathrm{MAX}}}$ **= 50**					$\boldsymbol{P^{\mathrm{MAX}}}$ **= 100**							
$\sqrt{\mathrm{k}}\sigma$	k	μ_{package}	$\mathrm{CV}_{package}$	DCL	HDP	APM	μ_{package}	$CV_{package}$	DCL	HDP	APM	μ_{package}	$\mathrm{CV}_{package}$	DCL	HDP	APM	μ_{package}	$\mathrm{CV}_{package}$	DCL	HDP	APM	μ_{package}	$\mathrm{CV}_{package}$	APM
	2	499.92	1.1362	0.00	0.0026	9.00	499.67	0.8125	0.00	0.00	13.75	499.70	0.8065	0.00	0.00	17.52	499.77	0.7864	0.00	0.00	24.49	499.99	0.1558	888.84
	3	499.98	0.1680	0.00	0.00	6.64	499.99	0.0740	0.00	0.00	7.16	500.00	0.0600	0.00	0.00	7.37	499.99	0.0460	0.00	0.00	7.72	500.00	0.0105	23.95
	4	499.99	0.0940	0.00	0.00	5.11	500.00	0.0500	0.00	0.00	5.27	500.00	0.0400	0.00	0.00	5.34	500.00	0.0300	0.00	0.00	5.45	500.00	0.0022	14.08
5	5	500.00	0.0680	0.00	0.00	4.15	500.00	0.0400	0.00	0.00	4.22	500.00	0.0320	0.00	0.00	4.27	499.99	0.0240	0.00	0.00	4.34	499.99	0.0008	10.46
	6	499.99	0.0540	0.00	0.00	3.45	499.99	0.0320	0.00	0.00	3.58	499.99	0.0260	0.00	0.00	3.63	499.99	0.0200	0.00	0.00	3.69	499.99	0.0004	8.22
	7	500.00	0.0500	0.00	0.00	3.06	500.00	0.0260	0.00	0.00	3.09	499.99	0.0220	0.00	0.00	3.11	499.99	0.0160	0.00	0.00	3.13	500.00	0.0003	6.70
	2	499.82	2.8370	0.00	0.0032	9.00	499.08	2.0257	0.00	0.00	13.66	499.36	1.9585	0.00	0.00	17.48	499.46	1.9141	0.00	0.00	24.14	500.02	0.4122	892.98
	3	499.96	0.3840	0.00	0.00	6.64	499.98	0.1860	0.00	0.00	7.17	500.00	0.1500	0.00	0.00	7.34	500.00	0.1160	0.00	0.00	7.66	500.00	0.0269	23.65
	4	499.99	0.2400	0.00	0.00	5.11	500.00	0.1280	0.00	0.00	5.27	499.99	0.1020	0.00	0.00	5.35	500.00	0.0780	0.00	0.00	5.45	499.99	0.0056	14.23
12.5	5	499.99	0.1740	0.00	0.00	4.14	499.99	0.1000	0.00	0.00	4.23	500.00	0.0800	0.00	0.00	4.27	500.00	0.0600	0.00	0.00	4.33	500.00	0.0019	10.38
	6	499.99	0.1360	0.00	0.00	3.45	499.99	0.0800	0.00	0.00	3.58	499.99	0.0660	0.00	0.00	3.63	499.99	0.0500	0.00	0.00	3.69	500.00	0.0010	8.12
	7	500.00	0.1260	0.00	0.00	3.06	499.99	0.0680	0.00	0.00	3.10	499.99	0.0540	0.00	0.00	3.11	500.00	0.0420	0.00	0.00	3.13	500.00	0.0007	6.54
	2	499.65	5.6760	0.00	0.0032	9.00	498.17	4.0589	0.00	0.00	13.66	498.72	3.9220	0.00	0.00	17.48	498.92	3.8323	0.00	0.00	24.14	500.06	0.8259	892.98
	3	499.88	0.8002	0.00	0.00	6.64	499.96	0.3720	0.00	0.00	7.15	500.01	0.3020	0.00	0.00	7.33	500.00	0.2320	0.00	0.00	7.68	500.00	0.0543	23.27
	4	499.98	0.4760	0.00	0.00	5.11	500.01	0.2580	0.00	0.00	5.27	500.01	0.2060	0.00	0.00	5.34	500.00	0.1560	0.00	0.00	5.45	500.00	0.0111	14.08
25	5	499.98	0.3540	0.00	0.00	4.14	499.99	0.2020	0.00	0.00	4.22	499.99	0.1600	0.00	0.00	4.26	500.00	0.1220	0.00	0.00	4.33	500.00	0.0040	10.30
	6	500.01	0.2760	0.00	0.00	3.45	500.00	0.1580	0.00	0.00	3.58	499.99	0.1320	0.00	0.00	3.63	499.99	0.1020	0.00	0.00	3.69	499.99	0.0020	8.14
	7	499.99	0.2580	0.00	0.00	3.06	499.99	0.1400	0.00	0.00	3.09	500.01	0.1120	0.00	0.00	3.11	499.99	0.0860	0.00	0.00	3.13	499.99	0.0014	6.67

It can also be observed that the average weight of the final package (μ_{package}) is not affected in any of cases studied. Additionally, note that the average maximum priority for each hopper (AMP) decreases as the k values increases, and at the same time, it is kept constant at different levels of CV (represented by $\sqrt{\text{k}}\sigma$).

Table 5 Trade-off values for maximum allowed priority (P^{MAX}) of 10, 30, 50 and 100

		Trade-off			
$\sqrt{\text{k}}\sigma$	k	$P^{\text{MAX}} = 10$	$P^{\text{MAX}} = 30$	$P^{\text{MAX}} = 50$	$P^{\text{MAX}} = 100$
5	2	0.00557	0.00375	0.00373	0.00365
	3	0.04548	0.01890	0.01492	0.01092
	4	0.05117	0.02713	0.02162	0.01611
	5	0.05325	0.03141	0.02520	0.01895
	6	0.05617	0.03404	0.02788	0.02162
	7	0.06830	0.03562	0.03025	0.02202
12.5	2	0.01371	0.00915	0.00882	0.00863
	3	0.10497	0.04828	0.03774	0.02787
	4	0.12848	0.06828	0.05425	0.04120
	5	0.13786	0.07972	0.06387	0.04798
	6	0.14452	0.08698	0.07236	0.05528
	7	0.18003	0.09782	0.07769	0.06055
25	2	0.02741	0.01830	0.01762	0.01725
	3	0.22420	0.09854	0.07769	0.05699
	4	0.25915	0.14013	0.11151	0.08396
	5	0.28413	0.16287	0.12918	0.09887
	6	0.29207	0.17101	0.14408	0.11231
	7	0.35543	0.19361	0.15537	0.11953

In table 5, the *Trade-off* values show that the largest increase in variability for each reduced unit of the priority occurs when $P^{\text{MAX}} = 10$, and they are the largest when we want to combine the greatest number of weights, i.e., when $k = 7$ (In the latter case, regardless of the value of P^{MAX}). In addition, the values for the *Trade-off* progressively decrease as the P^{MAX} value increases and they also progressively increase as the k value increases. Additionally, it is clear that the increases in the coefficient of variation (CV) cause increments in the *Trade-off* values.

5 Conclusions

A multihead weighing process is a packaging technology that can be of strategic importance to a company, as it can be a key to competitive advantage in the modern food industry. The improvement of the process quality and sensory quality of food packaged in a multihead weighers process investigated in this paper is relevant to industrial engineering. A bi-objective algorithm for the packaging processes in multihead weighers with an unequal supply is developed, and numerical experiments are provided to illustrate the performance of the proposed algorithm.

Our algorithm simultaneously deals with the objective of minimizing the difference between the target and the real package weight and the objective of maximizing the total priority of the chosen combination of hoppers. We propose using a single weighted performance or utility function that combines information about the two objectives or criteria being considered in this study (weight and priority), where the relative weight or importance of each objective is dynamically adjusted in each iteration or packing operation. Pareto-optimal solutions were obtained for both problems for the conflicting relationships between the objectives. To the best of our knowledge, no prior research has considered both objectives (weight and priority) to be potentially equally important.

The numerical experiments show that our algorithm succeeds in managing both objectives in a reasonable and efficient way. More precisely, the average highest observed priority (AMP) is significantly reduced compared to the mono-objective approach (in which weight is the only selection criterion), especially when the number k of hoppers to be combined is small, with average distance to target weight still remaining acceptable in general.

Nevertheless, it should be mentioned that the ratio between the increase in package dispersion and the improvement in terms of priority reduction (trade-off) remains within reasonable levels in all the tested cases. In relative (trade-off) terms, the largest effort in reducing priority occurs for higher values of k (which means more possible combinations) and smaller values of P^{MAX} (that is, as the priority objective becomes more important). Our proposal goes in a similar direction to the one by Karuno et al. (2007), but it offers a different insight into the packaging problem, as has been highlighted previously in the text. In conclusion, the effectiveness and efficiency of our approach has been shown.

Finally, the proposed method provides Pareto optimal sets of solutions to the problem of bi-objective optimization that can be analysed to obtain optimal configurations. The model will be useful to engineers concerned with the optimal configuration of a multihead weigher.

References

Barreiro, J.J., González, C. and Salicrú, M. (1998) ‘Optimization of Multiweighing Packing Proceeding’, *Top*, Vol. 6, No.1, pp. 37-44.

Branke, J., Deb, K., Miettinen, K. and Slowiński, R. (Eds.), (2008) *Multiobjective Optimization: Interactive and Evolutionary Approaches*, Springer: Berlin.

Collette, Y. and Siarry, P. (2013) *Multiobjective optimization: principles and case studies*, Springer Science & Business Media.

Ehrgott, M. (2005) Multicriteria optimization, 2nd ed., Springer; Berlin.

Garey, M.R. and Johnson, D.S. (1979) Computers and Intractability: A guide to the Theory of NP-Completeness, WH Freeman and Company, New York.

Imahori, S., Karuno, Y., Nagamochi, H. and Wang, X. (2011) ‘Kansei engineering humans and computers: Efficient dynamic programming algorithms for combinatorial food packing problems’, *International Journal of Biometrics*, Vol. 3, No.3, pp. 228-245.

Imahori, S., Karuno, Y., Nishizaki, R. and Yoshimoto, Y. (2012), ‘Duplex and Quasi-Duplex Operations in Automated Food Packing Systems’ *in 2012 IEEE/SICE International Symposium on System Integration (SII)*, Fukuoka, Japan, pp. 810-815.

Jaimes, A.L. and Coello Coello, C.A. (2008) ‘Multi-Objective Evolutionary Algorithms: A Review of the State-of-the-Art and some of their Applications in Chemical Engineering’ in Rangaiah, G.P. (Ed.), *Multi-Objective Optimization. Techniques and Applications in Chemical Engineering*, World Scientific, pp. 61-90.

Karuno, Y., Nagamochi, H. and Wang, X. (2010) 'Optimization Problems and Algorithms in Double-layered Food Packing Systems', *Journal of Advanced Mechanical Design, System, and Manufacturing*, Vol. 4, No.3, pp. 605-615.

Karuno, Y., Nagamochi, H. and Wang, X. (2007) 'Bi-criteria food packing by dynamic programming', *Journal of the Operations Research Society of Japan*, Vol. 50, No.4, pp. 376-389.

Karuno, Y., Takahashi, K. and Yamada, A. (2013) 'Dynamic Programming Algorithms with data rounding for combinatorial food packing problems', *Journal of Advanced Mechanical Design, System, and Manufacturing*, Vol. 7, No.2, pp. 233-243.

Karuno, Y. and Tateishi, K. (2014), 'Improved Heuristics with data rounding for combinatorial food packing problems' in *IEEE 2014 7th international conference on service-oriented computing and applications*, Matsue, Japan, pp. 81-88.

Keraita, J.N. and Kim, K-H. (2006) 'A Study on the optimum scheme for Determination of Operation time of Line Feeders in Automatic Combination Weighers', *Journal of Mechanical Science and Technology*, Vol. 20, No.10, pp. 1567-1575.

Keraita, J.N. and Kim, K-H. (2007) 'A Weighing Algorithm for Multihead Weighers', *International Journal of Precision Engineering and Manufacturing*, Vol. 8, No.1, pp. 21-26.

Liu, S. and Papageorgiou, L.G. (2013) 'Multiobjective optimisation of production, distribution and capacity planning of global supply chains in the process industry', *Omega*, Vol. 41, No. 2, pp. 369–382.

Marler, R.T. and Arora, J.S. (2004) 'Survey of multi-objective optimization methods for engineering', *Structural and Multidisciplinary Optimization*, Vol. 26, No.6, pp. 369-395.

Marler, T. (2009) *Multi-Objective Optimization: Concepts and Methods for Engineering*, VDM Verlag.

Montgomery, D.C. (2012) Design and Analysis of Experiments, 8th ed., John Wiley & Sons, New York.

Michalewicz, Z. and Fogel, D.B. (2004) *How to Solve It: Modern Heuristics*, 2nd ed., Springer Science & Business Media.

Pulido-Rojano, A. and García-Díaz, J.C. (2016), 'Analysis of the Filling Setting in the Multihead Weighing Process' *in Proceedings of the International Joint Conference - CIO-ICIEOM-IIE-AIM (IJC 2016)*, San Sebastián, Spain, pp. 521 - 528.

Rangaiah, G.P. and Bonilla-Petriciolet, A. (Eds.), (2013) *Multi-Objective Optimization in Chemical Engineering: Developments and Applications*, John Wiley & Sons.

Salicrú, M., González, C. and Barreiro, J.J. (1996) 'Variability Reduction with Multiweighing Proceedings', *Top*, Vol. 4, No.2, pp. 319-329.

Seng, C.K. and Rangaiah, G.P. (2009) 'Multi-objective optimization in food engineering', in Erdogdu F. (Ed.), *Optimization in Food Engineering*, Taylor and Francis/CRC Press, pp. 153- 176.

Yu, P.L. (1973) 'A class of solutions for group decision problems', *Management Science*, Vol. 19, No.8, pp. 936-946.

Zavala, G.R., Nebro, A.J., Luna, F. and Coello Coello, C.A. (2014) 'A survey of multi-objective metaheuristics applied to structural optimization'. *Structural and Multidisciplinary Optimization*, Vol. 49, No. 4, pp. 537–558.

Zeleny, M. (1973) 'Compromise programming', in Cochrane, J.L. and Zeleny, M. (Eds), *Multiple criteria decision making*, University of South Carolina Press, Columbia, pp. 262-301.