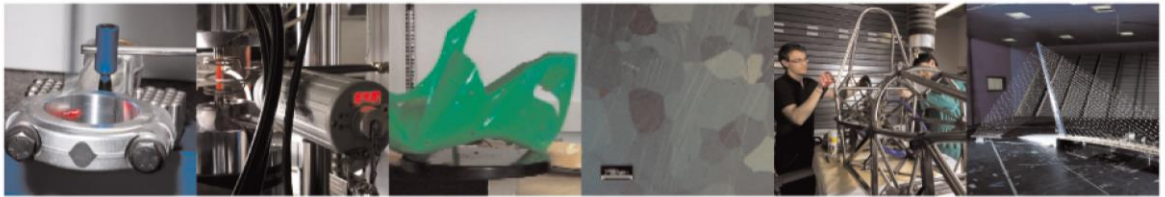




**POLITECNICO**  
MILANO 1863

DIPARTIMENTO DI MECCANICA



## Optimal setup of a multihead weighing machine

del Castillo, Enrique; Beretta, Alessia; Semeraro, Quirico

This is a pre-print version of an article published in EUROPEAN JOURNAL OF OPERATIONAL RESEARCH. The final authenticated version is available online at:

<http://dx.doi.org/10.1016/j.ejor.2016.10.017>

This content is provided under [CC BY-NC-ND 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/) license



# Optimal Targets Setup of a Multihead Weighing Machine

Enrique del Castillo\*

Department of Industrial and Manufacturing Engineering  
The Pennsylvania State University, University Park, PA 16802, USA

Alessia Beretta<sup>†</sup> and Quirico Semeraro<sup>‡</sup>

Dipartimento di Meccanica  
Politecnico di Milano, 20133 Milano, Italy

July 29, 2016

## Abstract

Multihead weighing machines (MWMs) are ubiquitous in industry for fast and accurate packaging of a wide variety of foods and vegetables, small hardware items and office supplies. A MWM consists of a system of multiple hoppers that are filled with product which when discharged through a funnel fills a package to a desired weight. Operating this machine requires first to specify the product weight targets or setpoints that each hopper should contain on average in each cycle, which do not need to be identical. The selection of these setpoints has a major impact on the performance of an MWM. Each cycle, the machine fills a package running a built-in knapsack algorithm that opens –or leaves shut– different combinations of hoppers releasing their content such that the total weight of each package is near to its target, minimizing the amount of product “given away”. In this paper, we address the practical open problem for industry of how to determine the setpoint weights for each of the hoppers *before* starting up the MWM, given a desired total package weight. An order statistic formulation based on a characterization of near-optimal solutions is presented. This is shown to be computationally intractable, and a faster heuristic that utilizes a lower bound approximation of the expected smallest order statistic is proposed instead. The setup solutions obtained with the proposed methods can result in substantial savings for MWM users. Alternatively, the analysis presented could be used by management to justify the acquisition of new MWM machines.

---

\*Corresponding author. Dr. Castillo is Distinguished Professor of Industrial & Manufacturing Engineering and Professor of Statistics. e-mail: exd13@psu.edu

<sup>†</sup>Ms. Beretta is a Ph.D. student in Mechanical Engineering.

<sup>‡</sup>Dr. Semeraro is a Professor in the Mechanical Engineering Department.

1  
2  
3  
4 Keywords: Combinatorial Weighers, Order Statistics, Packaging Technology, Multivariate Statistics.  
5  
6  
7

## 9 1 Introduction

10  
11  
12 A multihead weighing machine (hereafter an MWM, sometimes called a combinatorial  
13 weighing machine) is a computer-controlled machine used to fill a package with small  
14 products or parts with a given target weight. This machine has a wide range of applications  
15 in the food industry for packaging pasta, coffee beans, cereals, snacks, candies, vegetables,  
16 and even for packing poultry pieces and beef. Its applications cover also the packaging of  
17 non-food items, for instance, clips, nails, screws and a variety of other small hardware items.  
18 Among the multihead weigher manufacturers, the one with the world leading position has  
19 31,000 MWMs installed all over the world [12]. Despite their widespread use, analytical  
20 studies aimed at optimally setting up an MWM, a critical step affecting the performance  
21 of these machines, are lacking. In this paper, we model and analyze an MWM and propose  
22 methods for its optimal setup.  
23  
24  
25

26 A MWM is composed of a system of feeders, a set of  $H$  pool hoppers, a set of  $H$   
27 weight hoppers and a discharge chute to the packaging machine (Figure 1). The product  
28 is continuously fed via a central dispersion feeder (usually a vibrating cone) and  $H$  radial  
29 feeders (vibrating channels) to the pool hoppers. The role of the pool hoppers is to stabilize  
30 the product before dropping it into the weight hoppers. The average weight of product  
31  $\mu_i$ ,  $i = 1, \dots, H$ , that each hopper should contain must be specified by an operator before  
32 starting the machine. These average weights need not be identical. Once the machine is  
33 started, each cycle, a built-in knapsack-like algorithm selects a subset of hoppers whose  
34 sum of observed weights is closest to the target value after which a computer opens the  
35 selected hoppers releasing the product through the discharge chute into the package. Some  
36 hoppers can therefore remain shut filled with product from cycle to cycle. One cycle is  
37 repeated for each package. The performance of an MWM heavily depends on the initial  
38 hopper weights  $\{\mu_i\}$ . In industrial practice, operators currently use trial and error rules  
39 to setup the hopper weights based on the product to pack and the target weight of the  
40 package, but such setting-up operation may be far from optimal. In this paper, we focus  
41 on the analysis and optimal setup of MWMs with a single layer of hoppers (Figure 1), the  
42 most common type of MWM in industrial use.  
43  
44  
45  
46  
47

48 Practically all of the extant technical literature related to MWMs (see, e.g., [7, 8,  
49 9, 10, 11]), which mostly originates in Japan where MWMs were first developed, deals  
50 with the repetitive problem of finding the best combination of hoppers to open in each  
51 cycle, proposing different versions of Knapsack formulations, but does not address the  
52 *setup* problem of selecting the hopper weights *before* starting up the machine. The MWM  
53 problem we address below is somewhat related to canning problems [2, 13] but they differ  
54 in that they deal with a single target filling setting problem, and more importantly, there  
55 is no selection combination problem involved.  
56  
57

58 MWM's are based on an empirically observed "variance reduction" technique: it was  
59 noted that by filling a package from the combination of product from several hoppers,  
60 negative correlations are induced between the weights of product in opened hoppers given  
61  
62  
63  
64  
65

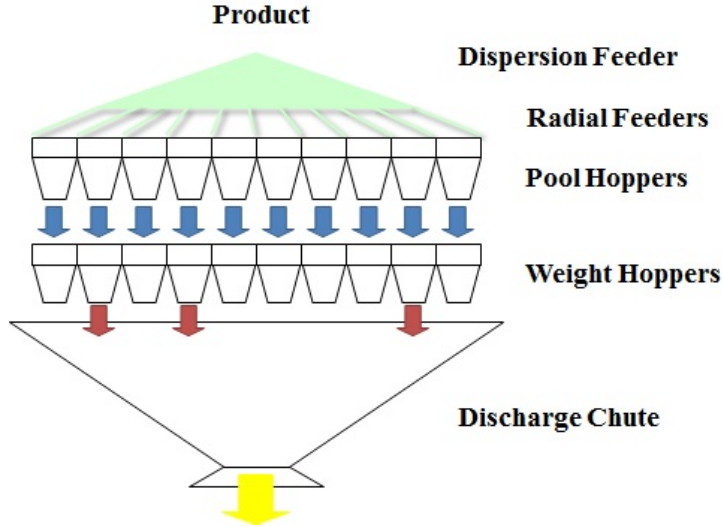


Figure 1: A single-layered multihead weighing machine. The central dispersion cone is connected to a series of vibrating radial feeders, one per hopper, which can be controlled individually providing individual controllability to each hopper mean weight setpoint  $\mu_j$ .

that they are random variables that are selected in each cycle subject to a constraint in their sum (which gives the package total weight) [8]. The negative correlations reduce the mean square error of the packages weight, “giving away” less product while satisfying the target constraint.

The rest of the paper is organized as follows. The next section presents a mathematical formulation of the MWM setup problem and an exact approach for simple problems (i.e. when only few combinations of hoppers opening are considered). Next, the behavior of good solutions obtained by numerical search is characterized. These characteristics are then used in section 4 to develop a heuristic approach to the optimal MWM setup problem. The paper ends with recommendations and directions for further research.

## 2 Formulation of the optimal setup problem of a multihead weigher machine

Let  $w_j$  be the observed weight of the product contained in the  $j$ th hopper in a particular cycle of operation,  $j = 1, 2, \dots, H$  where  $H$  is the number of hoppers in the machine. Each cycle the machine fills up a package with product released from a subset of the hoppers and the depleted hoppers are refilled. Assume  $w_j$  is a realization of the random weight  $W_j \sim N(\mu_j, \sigma_j^2 = \alpha^2 \mu_j^2)$ ,  $\mu_j > 0, j = 1, 2, \dots, H$  and assume each weight is independent of other weights  $W_i (i \neq j)$ . The proportionality constant  $\alpha$  (with  $\alpha < 1$ ) is assumed known and given as it depends on the product to be packed. The proportional relation between mean and standard deviation of the weights is known to exist in this type of machines (e.g., see [8]). We point out that as long as  $\sigma = f(\mu)$  holds for any known  $f$ , the methods developed below also apply after trivial modification. However, given that the available

empirical evidence (see, e.g. [3], p. 87) indicates that a simple linear relation of the form  $\sigma = \alpha\mu$  fits the weight data very well, it was adopted in what follows. A *setup* of the machine consists of specifying the values of the setpoints  $\boldsymbol{\mu}' = (\mu_1, \mu_2, \dots, \mu_H)$ , to which, according to our assumption, also determine the hopper weight variances  $\sigma_j^2, j = 1, \dots, H$ , for a given target value  $T$  that specifies the minimum weight content of each package to be filled. Once the machine is setup, the combinatorial weigher machine starts to fill packages of product, solving a knapsack algorithm per package. Our goal is to determine the best setpoints  $\boldsymbol{\mu}$  according to some specific criteria on the weight content of the packages.

While there are different knapsack formulations that have been reported in the MWM literature, most of them utilize a linear objective function and linear constraints. In this section, we assume the machine has a built-in algorithm that solves for each package the deterministic knapsack problem:

$$\min \quad \omega_{\bullet} = \sum_{j=1}^H \delta_j w_j \quad \text{subject to:} \quad \omega_{\bullet} = \sum_{j=1}^H \delta_j w_j \geq T \quad (1)$$

where  $\delta_j$  is either 0 or 1. In this formulation, the total observed package weight  $\omega_{\bullet}$  is required to be as small as possible but larger or equal to the given target package weight  $T$ . Prior to observing the hopper weights  $\{W_j = w_j\}$  in any cycle, the total package weight  $\mathcal{W}_{\bullet}$  is the minimum of  $K$  dependent, not identical normal random variables  $X_i$  for  $i = 1, 2, \dots, K$ , subject to the constraint  $\mathcal{W}_{\bullet} \geq T$ , where  $K$  equals the total number of possible combinations of opened/closed hoppers from which the knapsack algorithm can select (choosing the  $\delta_j$  variables above). We hasten to point out that *we are not concerned with solving the knapsack problem; the knapsack problem is internal to the machine and considered given*. We are concerned with determining the setpoints of the machine, i.e., the mean weights in each hopper, which are the “inputs” of the system as depicted in Figure 2.

The optimal setpoints could be found from the distribution of the optimum objective function value (i.e., the package weight  $\mathcal{W}_{\bullet}$ ) of a random Knapsack where  $W_j$  substitutes  $w_j$  in (1). However, there are only limited results related to this distribution ([15], p. 526). They are asymptotic results as the number of hoppers  $H \rightarrow \infty$  under the assumption the hopper weights  $W_i$ 's are  $U(0, 1)$  random variables, which is clearly not our case. In the remainder of this section we describe how to compute the exact *moments* of the total weight package  $\mathcal{W}_{\bullet}$  in problem (1) and how this leads very rapidly to computational complexities in practice.

If all possible combinations of any number of hoppers can be selected to open (or close) in a cycle, then clearly there are  $K = \sum_{i=0}^{H-1} \binom{H}{i} = 2^H - 1$  combinations, since we can assume that at least one hopper will open in each cycle to let some product get in the package. In later sections,  $K$  will denote the number of combinations that leave up to  $s < H$  hoppers shut in a cycle, in which case  $K = \sum_{i=0}^s \binom{H}{i}$ . From now on, we use the term *combination* to refer to a specific selection of  $\delta_j, j = 1, 2, \dots, H$  variables in (1), i.e., to a specific selection of hoppers that are opened in a cycle. We also apply this term by extension to the package weight  $X_i = \sum_{j=1}^H \delta_j W_j$  that combination  $i$  generates ( $i = 1, \dots, K$ ).

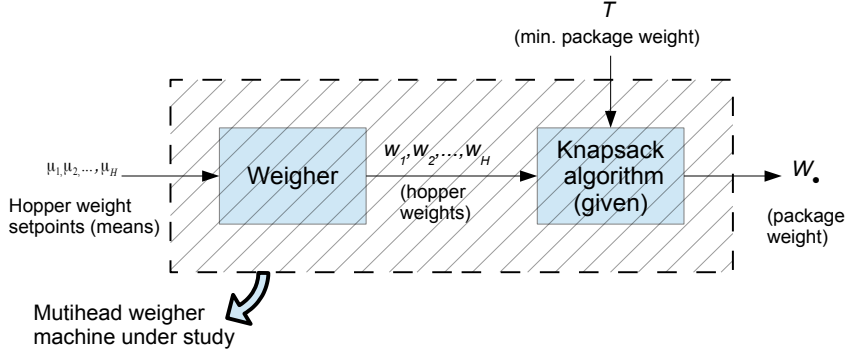


Figure 2: Information flow in the system under study. The setup problem in a multihead weigher machine consists in finding values of the hopper weights  $\mu_1, \dots, \mu_H$  that optimize some property of the resulting package weight  $\omega_\bullet$ . The built-in knapsack algorithm the machine comes equipped with is considered internal and given. Therefore, the system under study (delimited by dashed lines and hatched) requires determination of the “inputs” (the  $\mu_i$ ’s) to optimize some property of the “output”  $\omega_\bullet$ .

To consider the different combinations of weights, let matrix  $\mathbf{P} = \{p_{ij}\}$  be a  $K \times H$  matrix where  $\{p_{ij} = \delta_j|i\}$ , with  $p_{ij} = 1$  if combination  $i$  includes opening hopper  $j$  and  $p_{ij} = 0$  if otherwise,  $i = 1, \dots, K$ . We then define the random vector of combinations  $\mathbf{X}' = (X_1, X_2, \dots, X_K)$  as:

$$\mathbf{X} = \mathbf{P}\mathbf{W} \quad (2)$$

where  $\mathbf{W}' = (W_1, W_2, \dots, W_H)$  follows a multivariate  $N(\boldsymbol{\mu}, \Sigma_W)$  distribution with  $\Sigma_W = \text{diag}(\alpha^2 \mu_j^2)$ . It follows that for the different weight combinations that can be formed we have that:

$$\mathbf{X} \sim N(\boldsymbol{\Theta}, \Sigma) \quad \text{where} \quad \boldsymbol{\Theta} = E[\mathbf{X}] = \mathbf{P}\boldsymbol{\mu}, \quad \text{and} \quad \Sigma = \text{Var}(\mathbf{X}) = \mathbf{P}\Sigma_W\mathbf{P}'.$$

The  $K \times K$  matrix  $\Sigma$  includes the covariances between the random weights resulting from the different combinations of selected hoppers, some of which may be large, depending on the  $K$  combinations to consider. If combinations that “share” many hoppers are included,  $\Sigma$  may be close to rank deficient.

**Problem definition.** The optimal MWM setup problem we address, that corresponds to the knapsack problem (1) requires solving:

$$\min_{\boldsymbol{\mu}} \text{MSE}(\mathcal{W}_\bullet | \mathcal{W}_\bullet \geq T)$$

that is, finding the hopper setpoints such that the mean square error of the package weight  $\mathcal{W}_\bullet = \min(X_1, X_2, \dots, X_K)$  with  $\mathcal{W}_\bullet \geq T$  is minimized, since it is desired no package should weigh less than  $T$ . The distribution of  $\mathcal{W}_\bullet$  is a function of the hopper mean weights  $\boldsymbol{\mu}$ . Note that the  $\mu_j$  are not required to be integers. Therefore, we must first *find the distribution of a constrained smallest order statistic of a set of correlated normal variables*, and setup an optimization problem with it. As far as we know, there are no published results related to such problem in the Order Statistics literature. Afonja [1] found expressions for the first two moments of the *unconstrained* maximum order statistic of a set of correlated normals. The constraint  $\mathcal{W}_\bullet \geq T$  considerably increases the computational complexity when obtaining the moments in exact form.

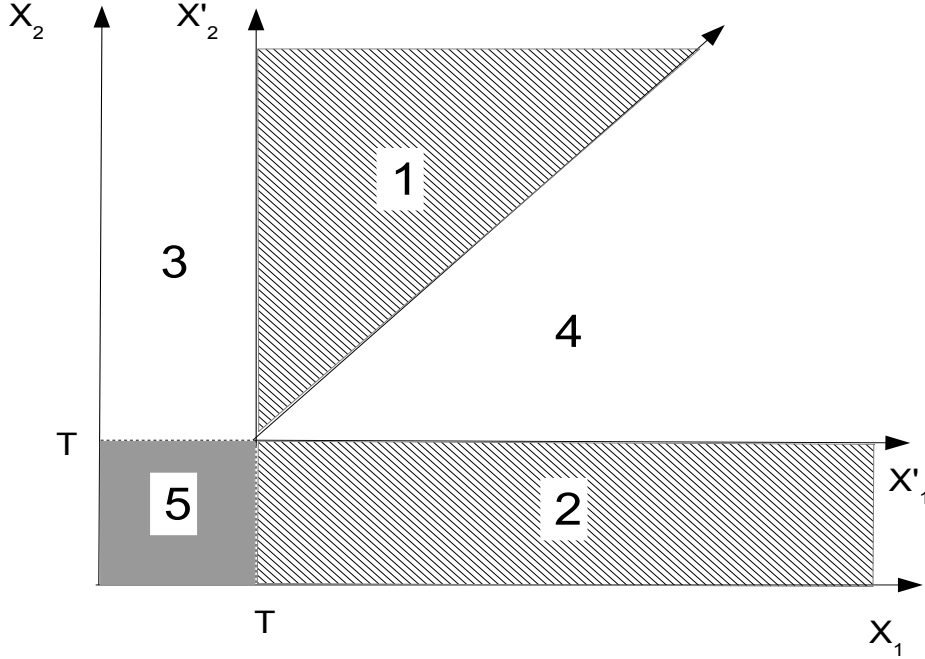


Figure 3: A “toy” weigher machine setup problem with two combinations. Hashed regions 1 and 2 are areas where combination  $X_1$  is better than combination  $X_2$  ( $X_1$  is closer to and greater than  $T$ ); in regions 3 and 4  $X_2$  is better ( $X_2$  is closer to and greater than  $T$ ). Shaded region 5 is infeasible. The search for a feasible solution needs to be undertaken over regions 1 through 4.

We show next that the computational expense of the expressions needed to obtain the moments of such constrained minimum is too large in general, motivating the approximate approach shown in a later section.

Consider first the simple case where there are only two combinations of hoppers ( $K = 2$ ). Figure 3 shows the regions where each random variable  $X_i$  ( $i = 1, 2$ ) achieves the minimum weight. The two hashed regions (numbers 1 and 2) are areas over which  $X_1$  is minimum and greater than  $T$ ; symmetrically, the two unshaded areas (3 and 4) are where combination  $X_2$  is the minimum and greater than  $T$ . We point out that we do not seek to find  $\min(X_1|X_1 \geq T, X_2|X_2 \geq T)$ , which would correspond only to taking the minimum over areas 1 and 4 in Figure 3. This is not what we seek, since, for instance, we could have  $X_1 \geq T$  but  $X_2 < T$  and still have found a feasible solution to our packing problem; all we need is at least one combination to be larger or equal to  $T$ . Thus, for  $H = 2$  we need to search in all four areas 1 to 4 in Figure 3. If we define  $\mathcal{W}_\bullet = \min(X_1, X_2)$ , then for a feasible solution to exist this minimum must be greater or equal to the target  $T$ , i.e.,  $\mathcal{W}_\bullet \geq T$ . Note that *the minimum of the combinations is what is constrained, not the individual combination weights.*

Define  $X'_i = X_i - T$ , as in the figure. The weigher machine must select one of the variables  $X'_i$  to fill up a package. The  $r$ th moment of the selected weight  $\mathcal{W}_\bullet$  is therefore

given by:

$$\begin{aligned}
E[\mathcal{W}_\bullet^r | \mathcal{W}_\bullet \geq T] &= \int_0^\infty x_1^{r'} \int_{x_1'}^\infty \phi(\mathbf{X}') dx_1' dx_2' + \int_0^\infty x_1^{r'} \int_{-T}^0 \phi(\mathbf{X}') dx_1' dx_2' \\
&+ \int_0^\infty x_2^{r'} \int_{x_2'}^\infty \phi(\mathbf{X}') dx_2' dx_1' + \int_0^\infty x_2^{r'} \int_{-T}^0 \phi(\mathbf{X}') dx_2' dx_1' \quad (3)
\end{aligned}$$

where  $\phi(\mathbf{X}')$  is the (bivariate) normal density of  $\mathbf{X}'$ , i.e.,  $N(\boldsymbol{\Theta} - \mathbf{T}, \boldsymbol{\Sigma})$  with  $\mathbf{T}$  equal to a  $K$ -vector filled with the package target weight  $T$ . The four terms correspond to integrals over the probability measure in areas 1, 2, 4 and 3 in Figure 3, respectively. Solving the setup problem for a combinatorial machine in this case consists in minimizing  $\text{MSE}(\mathcal{W}_\bullet | \mathcal{W}_\bullet \geq T) = \text{Var}(\mathcal{W}_\bullet | \mathcal{W}_\bullet \geq T) + (E[\mathcal{W}_\bullet | \mathcal{W}_\bullet \geq T] - T)^2 = E[\mathcal{W}_\bullet^2 | \mathcal{W}_\bullet \geq T] - E[\mathcal{W}_\bullet | \mathcal{W}_\bullet \geq T]^2 + (E[\mathcal{W}_\bullet | \mathcal{W}_\bullet \geq T] - T)^2$  with respect to  $\mu_1$  and  $\mu_2$ . Therefore, the integrals above will need to be performed several times inside an optimization routine.

Consider next the case of  $K = 3$  different combinations the MWM can select from. In this case we will get more rectangular areas similar to regions 2 and 3 in Figure 3. Specifically, the  $r$ th moment of the selected weight is given by the expression:

$$\begin{aligned}
E[\mathcal{W}_\bullet^r | \mathcal{W}_\bullet > T] &= \sum_{i=1}^3 \left\{ \int_0^\infty x_i^{r'} \int_{x_i'}^\infty \int_{x_i'}^\infty \phi(\mathbf{X}') d\mathbf{x}' + \int_0^\infty x_i^{r'} \int_{-T}^0 \int_{-T}^0 \phi(\mathbf{X}') d\mathbf{x}' \right. \\
&+ \left. \int_0^\infty x_i^{r'} \int_{x_i'}^\infty \int_{-T}^0 \phi(\mathbf{X}') d\mathbf{x}' + \int_0^\infty x_i^{r'} \int_{-T}^0 \int_{x_i'}^\infty \phi(\mathbf{X}') d\mathbf{x}' \right\} \quad (4)
\end{aligned}$$

which is an expression with twelve 3-dimensional integrals where

$$12 = \left[ \binom{K-1}{K-1} + \binom{K-1}{K-2} + \dots + \binom{K-1}{0} \right] \cdot K = 2^{K-1} \cdot K$$

for  $K = 3$ .

Evidently,  $2^{K-1} \cdot K$  grows very fast. If *all* the combinations of  $H$  hoppers are considered (so  $K = 2^H - 1$ ) the total number of  $2^H - 1$ -dimensional integrals,  $2^{2^H-2} \cdot (2^H - 1)$ , grows extraordinarily fast, see Table 1.

No. of hoppers ( $H$ )	No. of Integrals	Dimension of each integral ( $K$ )
1	1	1
2	12	3
3	448	7
4	245760	15
5	3.3286 e10	31
8	7.3817 e78	255

Table 1: Number of multidimensional integrals needed to compute a moment  $E[\mathcal{W}_\bullet^r | \mathcal{W}_\bullet \geq T]$  of the package weight assuming all  $2^H - 1$  possible combinations are considered.

Although in practice not all possible combinations need to be considered, the exponentially increasing number of combinations and the required computations render an exact



approach intractable for problems of realistic size. Therefore, we proceed next to develop approximate approaches to the solution of this problem. We first characterize the properties of near-optimal solutions obtained by a search procedure in order to seek similarly good solutions in a heuristic approach to be discussed in section 4.

### 3 Characterization of near optimal solutions to the MWM setup problem

As mentioned earlier, hereon by a ‘good’ solution we mean a low (or near minimum) MSE solution subject to the constraint  $\mathcal{W}_\bullet \geq T$ . For computational simplicity, in this section we only consider simulated MWMs with  $H = 4$  and  $H = 5$  hoppers when none, one, or two hoppers can remain shut in a cycle( i.e.,  $s = 2$ ). The target value  $T = 500$  and the parameter  $\alpha = 0.123$  were used throughout. There is no loss of generality since the characteristics discussed in this section are independent of these parameters. Figure 4

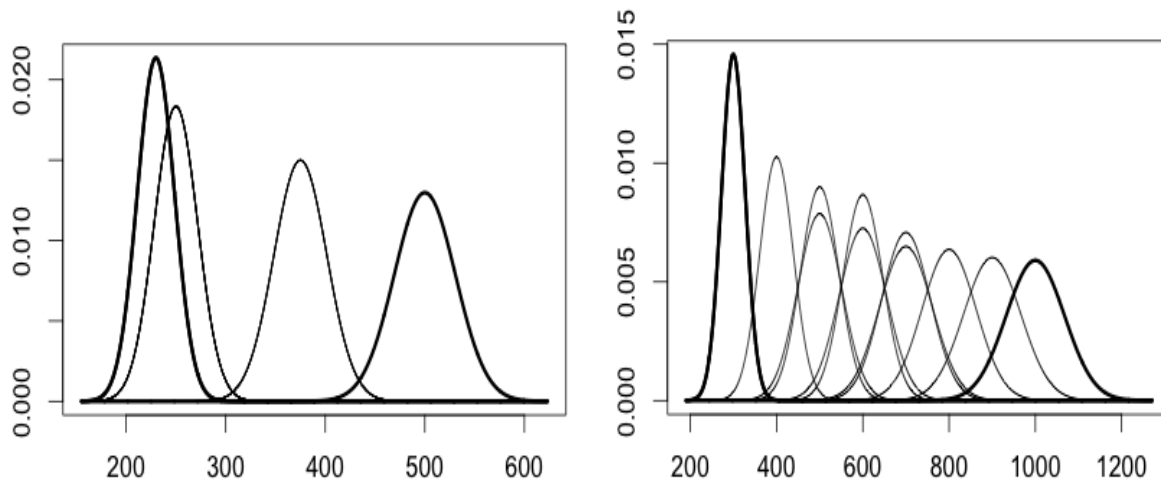


Figure 4: Marginal density plots of the combination weights  $\{X_i\}$  generated by two non-optimal solutions of a weigher problem with  $H = 4$  hoppers. Plot on the left is for the solution  $\mu' = (500/4, 500/4, 500/4, 500/4)$ . Plot on the right is for the solution  $\mu' = (400, 300, 200, 100)$ . Combinations that leave up to  $s = 2$  closed hoppers were considered. Darker density lines correspond to  $X_{[1]}$  and  $X_{[K]}$ .

shows the marginal densities of the  $K = 11$  combinations  $X_i, i = 1 \dots, 11$ , generated in a  $H = 4$  hopper problem for two typical, but not optimal solutions. The marginal densities of the first order statistic  $X_{[1]}$  and the last order statistic  $X_{[K]}$  (the moments of these densities were obtained as described in Appendix A) are highlighted. The plot on the left corresponds to the setpoints  $\mu' = (T/4, T/4, T/4, T/4)$ , a ‘‘logical’’ solution in which all hopper setpoint weights are equal. As it can be seen, the weigher machine will have only one combination available to fill the  $T = 500$  gr. packages, namely, the density of the largest order statistic which corresponds to all hoppers opened in each cycle. Note also how there are many densities that overlap perfectly, since there are  $K = 11$  combinations,

but only three are uniquely different densities. This is a very poor solution, since there will be about a 50 % chance that the constraint  $\mathcal{W}_\bullet \geq T = 500$  will not be satisfied. The graph on the right shows the marginal densities for a different ‘ladder’ solution with setpoints  $\mu' = (400, 300, 200, 100)$ . The marginal densities are much more dispersed, which is better than the previous solution. But note how the densities are not dispersed symmetrically around  $T = 500$ , and in particular, note how there are only a few ‘good’ combinations, i.e., densities located near or above  $T = 500$ , so the performance will be far from optimal.

In contrast with the two previous solutions which can be classified as poor, Figure 5 shows the marginal densities for near-optimal solutions for the  $H = 4$  case and the  $H = 5$  cases, obtained by a simulation-based search of the best combination as reported in [4]. This search attempts to minimize the MSE of the selected weights  $\mathcal{W}_\bullet$  such that  $\mathcal{W}_\bullet \geq T$ . The solution found for the  $H = 4$  problem is  $\mu' = (294.5, 276.7, 183.7, 66.6)$ . For the  $H = 5$  hopper MWM there are  $K = 16$  combinations possible (with up to  $s = 2$  hoppers remaining shut), and the densities shown in the figure correspond to the solution  $\mu' = (203.7, 178.6, 110.9, 191.0, 55.7)$ . Some useful characteristics of the solutions  $\mu^*$  that

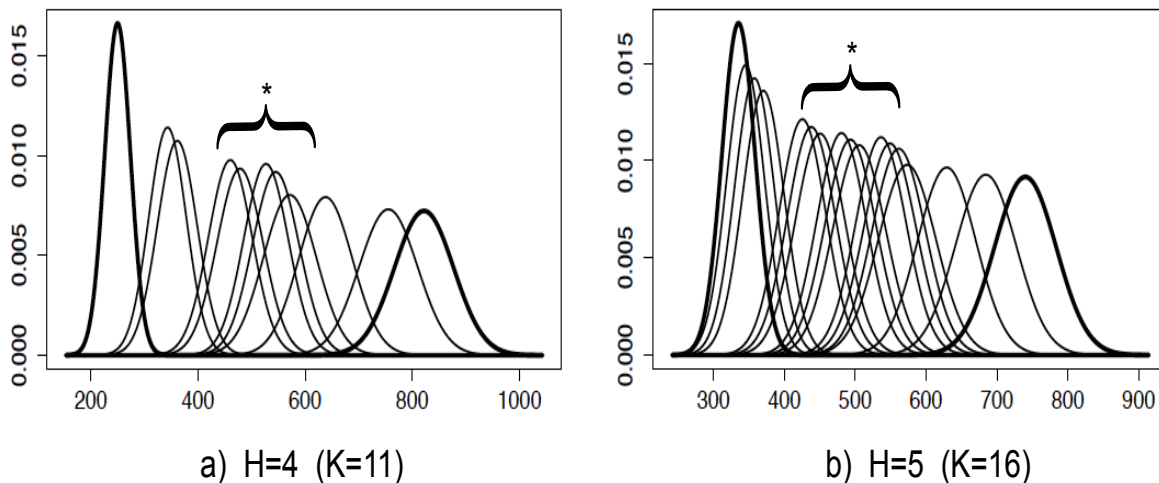


Figure 5: Density plots of the near-optimal  $\{X_i\}$  combination weights for a problem with a) 4 hoppers and b) 5 hoppers. Combinations that leave up to  $s = 2$  closed hoppers were considered. Darker density lines correspond to  $X_{[1]}$  and  $X_{[K]}$ . Note the cluster of densities (\*) around  $T = 500$  (see text).

generated the combinations shown in Figure 5 are shown in Table 2.

An interesting characteristic common to all near-optimal both solutions such as those shown in Table 2 is that the average of the two extreme order statistics,  $(E[X_{[1]}] + E[X_{[K]}])/2$  remains slightly above  $T$ . At the same time, the combinations are quite dispersed. This leads to our first empirical characterization of a good solution.

*Characteristic 1:* in a good solution, the combinations  $\{X_i\}_{i=1}^K$  should be such that the average of the two (unconstrained) extreme order statistics is larger than  $T$ .

Likewise, good solutions such as those in Table 2 have a cluster of *many* densities around  $T$ , indicated with a “\*” in Figure 5. These densities should not all be near identical, but should differ “enough”, thus some variability in them is desirable. How to define “somewhat

Property	$H = 4$	$H = 5$
$\mu^*$	(294.9, 276.7, 183.7, 66.6)	(203.7, 191.0, 178.6, 110.9, 55.7)
$(E[X_{[1]}] + E[X_{[K]}])/2$	536.0	537.7
$\bar{\Theta}$	523.2	508.7
$-\log(\det(\Sigma))$	49.67	89.42
$p \equiv \sum_{i=1}^K P\{X_i \in (T, T + 100)\}$	2.78	4.79
$c(\Theta, T)$	4	7
$-\log(\det(\mathbf{M}))$	26.10	66.19
$\widehat{E}(\mathcal{W}_\bullet   \mathcal{W}_\bullet > T)$	522.8	514.2
$\widehat{\text{Var}}(\mathcal{W}_\bullet   \mathcal{W}_\bullet > T)$	301.1	129.2
$\widehat{\text{MSE}}(\mathcal{W}_\bullet   \mathcal{W}_\bullet > T)$	822.7	331.1

Table 2: Some characteristics of the near optimal (min MSE) solutions for  $H = 4$  and  $H = 5$  hoppers,  $T = 500, \alpha = 0.123$  obtained by searching combinations with up to 2 hoppers shut. A heuristic method is developed to find solutions that mimic all these characteristics. Quantities listed below the line were estimated via simulation for the solution in question.

different” is difficult. In our numerical results, we obtained good performance by defining two solutions to be different enough if they differ by 4% or more of the target weight. We hence defined  $u(\Theta, T)$  to be the vector of values  $\Theta_i$  that differ by at least  $0.04T$  of other  $\Theta_i$ ’s:  $u(\Theta, T) \equiv \text{unique}(\Theta - (\Theta \bmod 0.04 \cdot T))$ , where we assume we have available a function `unique` that returns the different items in a vector. Then, we define the number of densities with significantly different locations in an interval around  $T$  ( $0.8 \cdot T, 1.2 \cdot T$ ) as:

$$c(\Theta, T) = \sum_i \{(u_i > 0.8 \cdot T) \text{ and } (u_i < 1.2 \cdot T)\}$$

These non-identical densities around  $T$  provide significantly different combinations in each cycle to the knapsack algorithm to select from and fill up a package. Hence, for a near-optimal solution we observe that the sum of the marginal probabilities  $\sum_{i=1}^K P\{X_i \in (T, 1.2 \cdot T)\}$  is large relative to a non-optimal solution. This can be summarized in:

*Characteristic 2:* good solutions generate combinations whose densities cluster around  $T$  and are characterized by a large count number of uniquely different densities  $c(\Theta, T)$  as defined above. Thanks to this cluster of densities, good solutions are associated to a relatively large value of  $p(\Theta, T) \equiv \sum_{i=1}^K P\{X_i \in (T, 1.2 \cdot T)\}$ , the probability of package weights just above  $T$ , compared to non-optimal solutions.

The marginal density functions of each combination, however, are usually positively correlated and the probability  $p(\Theta, T)$  is not very informative by itself unless the correlations are accounted for. *A set of highly positively correlated combinations behaves essentially as fewer combinations.* The determinant of  $\Sigma$  is a simple measure of the global degree of correlation in the combinations. It was observed that the values of  $\log(\det(\Sigma))$  are relatively large for the near-optimal solutions compared to non-optimal solutions. This implies that in a good solution, the resulting combinations  $\{X_i\}_{i=1}^K$  are relatively less correlated. Thus, the densities should be as little correlated as possible (especially those close to  $T$ ), in order to provide as uncorrelated combinations as possible to the knapsack algorithm. Apart of

the cluster of good combinations close to  $T$ , other densities should disperse in a symmetrical manner around  $T$ . To emphasize these two aspects, low correlation of the combinations located near  $T$  and variability of their locations around  $T$ , we define the matrix:

$$\mathbf{M} = \mathbf{\Sigma} + (\mathbf{\Theta} - \mathbf{T})(\mathbf{\Theta} - \mathbf{T})'$$

which adds to  $\mathbf{\Sigma}$  a measure of variability of the locations of the densities around  $T$  in the second term. The values observed for  $\log(\det(\mathbf{M}))$  for the near-optimal solutions are large relative to non-optimal solutions (Table 2). This has the effect of dispersing the densities, avoiding too similar densities around  $T$ . In summary, we have the following.

*Characteristic 3.* A good solution provides large values of  $\det(\mathbf{M}) \equiv |\mathbf{M}|$  relative to non-optimal solutions.

Finally, we point out an obvious but useful property of any solution  $\boldsymbol{\mu}$ . The combinatorial weigher machine does not give preference to any hopper over the others, and each hopper can be set in exactly the same way as the others. In other words, we have:

*Characteristic 4.* The quality of a solution  $(\mu_1, \mu_2, \dots, \mu_H)$ , as measured by any function of  $\mathcal{W}_\bullet$  (e.g.  $\text{MSE}(\mathcal{W}_\bullet | \mathcal{W}_\bullet \geq T)$ ) is invariant to any permutation  $\sigma(\cdot)$  of the hoppers indices  $(\mu_{\sigma(1)}, \mu_{\sigma(2)}, \dots, \mu_{\sigma(H)})$ .

This is useful since we can reduce the search conducted by any optimization algorithm to the region  $\mu_1 \geq \mu_2 \geq \dots \geq \mu_H$ . Assuming we try  $L$  integer values or “levels” of weights in each of the  $H$  hoppers, this obviously generates  $K = L^H$  combinations. However, it is easy to see that if only integer solutions are considered, and we try  $L$  levels in each hopper, the constraint  $\mu_1 \geq \mu_2 \dots \geq \mu_H$  reduces the combinations from  $K = L^H$  to  $K = (L + H - 1)! / (H!(L - 1)!)$ . While a considerable reduction, enumeration of these combinations is still not an option for realistic number of hoppers ( $H \geq 8$  is the norm). For instance, for  $H = 6$  and using a realistic value  $L = 100$  grams, the “ladder” constraint still calls for a search space consisting in 1.6E+09 combinations, which need to be evaluated somehow, via some objective function (like MSE, which is out of the question) or via simulation, which will be very time consuming.

As we will show in the next section, excellent solutions can be obtained by optimizing our objective function over a space that contains only combinations where up to  $s = 3$  hoppers are kept shut in a cycle. Our intuition for searching only combinations where a low number of hoppers remains shut in a cycle is that, if one has a MWM with a large number of hoppers, then the properties of the machine will be defeated if we open only a few hoppers each cycle. It is the *combined* action of many hoppers what makes the package contents approach the target weight  $T$  from the right and save considerable money to companies.

We now present a heuristic method that aims at producing solutions that meet the aforementioned four characteristics.

## 4 A heuristic optimization model for determining the MWM hopper setpoints

The four characteristics identified in the previous section as common to all near-optimal solutions can be used to devise a heuristic algorithm for setting up an MWM. We therefore propose to solve:

$$\text{maximize } \log|\mathbf{M}| + p(\Theta, T) + c(\Theta, T) \quad (5)$$

subject to:

$$\frac{E[X_{[1]}] + E[X_{[K]}]}{2} > 1.1 \cdot T \quad (6)$$

$$\mu_1 \geq \mu_2 \geq \dots \geq \mu_H \quad (7)$$

$$0 < \mu_i < f \cdot T \quad (8)$$

The log used in (5) is a standard way of scaling (up) a numerically small determinant. The upper limit in (8) is aimed to reduce the search space of each setpoint. It was observed that as  $H$  increases, the optimal values of the  $\mu_i$  decrease, so increasingly smaller values of  $f$  with  $f < 1$  should be used.

Finally, in addition to constraints (6)-(8) it is necessary to ensure that there will always be at least a feasible combination in a cycle, i.e., one combination, that in which all hoppers open, should generate more product than  $T$  with very high probability. Under the assumptions discussed in section 2, this can be expressed as:

$$\Phi \left( \frac{T - \sum_{i=1}^H \mu_i}{\alpha \sqrt{\sum_{i=1}^H \mu_i^2}} \right) \leq \epsilon$$

which, since  $\mu_i > 0$  can be written as:

$$\|\boldsymbol{\mu}\|_1 + \alpha \|\boldsymbol{\mu}\|_2 \Phi^{-1}(\epsilon) \geq T \quad (9)$$

The complete heuristic solution consists in maximizing (5) subject to constraints (6)-(9). To solve it, we use an Augmented Lagrange routine in R's `Nloptr` library (see Appendix B for details of our R implementation). Since the optimization problem is clearly non-convex, the optimizer was started from a grid of initial near-feasible trial points (equations (7), (8) and (9) are satisfied at the initial points, although (6) may not be satisfied). Table 3 shows results for  $H = 4$  and  $H = 5$  when the same combinations and parameters as for the solutions in Table 2 were used.

As can be seen, the solutions found with the heuristic approach are very close to those in Table 2, found by a simple search on a simulator of the machine. For  $H = 4$  the heuristic provides a slightly better MSE, whereas for  $H = 5$  the opposite occurs. Although characterizing the solutions increases the understanding of the problem, the heuristic method in this section is only computationally feasible when the number of combinations  $K$  considered is very small. We now present a simple modification of this heuristic which allows a user to tackle larger problems.

Property	$H = 4$	$H = 5$
$\boldsymbol{\mu}^*$	(267.4, 259, 234.6, 57.7)	(228.1, 200.3, 161.9, 113.5, 61.0)
$(E[X_{[1]}] + E[X_{[K]}])/2$	549.9	549.6
$\bar{\Theta}$	521.0	525.8
$-\log(\det(\boldsymbol{\Sigma}))$	49.79	89.07
$p \equiv \sum_{i=1}^K P\{X_i \in (T, T + 100)\}$	3.62	4.79
$c(\boldsymbol{\Theta}, T)$	6	9
$-\log(\det(\boldsymbol{M}))$	26.2	65.84
$\widehat{E}(\mathcal{W}_\bullet   \mathcal{W}_\bullet > T)$	520.7	514.6
$\widehat{\text{Var}}(\mathcal{W}_\bullet   \mathcal{W}_\bullet > T)$	340.3	132.6
$\widehat{\text{MSE}}(\mathcal{W}_\bullet   \mathcal{W}_\bullet > T)$	769.8*	347.0

Table 3: Solutions obtained with the heuristic method (5-9) for  $H = 4$  and  $H = 5$  hoppers,  $T = 500$ ,  $\alpha = 0.123$ ,  $\epsilon = 1e - 05$ ,  $f = 0.6$  ( $H = 4$ ) and  $f = 0.5$  ( $H = 5$ ), 100 initial trials (up to 2 hoppers can remain shut). Quantities listed below the line were estimated via simulation for the solution in question. Compare to Table 2 (\* indicates the heuristic gives better MSE solution). The exact expression for  $E[X_{[1]}]$  was used.

## 4.1 Modified optimization heuristic for faster solution

The main computational bottleneck of the heuristic optimization is the *exact* computation of the expected values of the extreme values in constraint (6). We observed that while  $E[X_{[K]}]$  could be well approximated with  $\max(\Theta_i)$ , a similar approximation is not possible for  $E[X_{[1]}]$ . As shown in Appendix A, computing this expectation requires  $K$   $K$ -dimensional Normal integrals, which can only be attempted for small  $K$ . Unfortunately, we found constraint (6) to be critical and hence it cannot be removed.

As an alternative, we could use a fast-to-compute lower bound  $LB(E[X_{[1]}])$  instead of the computationally expensive  $E[X_{[1]}]$  in constraint (6). Bertismas et al. [6] (Theorem 4) give a useful closed-form lower bound for  $E[X_{[1]}]$  in a collection of possibly correlated normals. This bound is given by:

$$\begin{aligned}
E[X_{[1]}] \geq LB(E[X_{[1]}]) \equiv & -\frac{1}{2} \left[ \sum_{i=1}^K \left( -\mu_i + \sqrt{\left( -\mu_i - \max_i \left\{ -\mu_i + \frac{K-2}{2\sqrt{K-1}} \sigma_i \right\} \right)^2 + \sigma_i^2} \right) \right] \\
& - \frac{2-K}{2} \max \left\{ -\mu_i + \frac{K-2}{2\sqrt{K-1}} \sigma_i \right\} \quad (10)
\end{aligned}$$

Bound (10) is extremely fast to compute compared to the exact moment, as it requires no integration. Table 4 shows some computation times and MSE values for comparison purposes when solving (5-9) using the exact moment and its lower bound approximation.

We have coded both versions of the heuristic, using the exact  $E[X_{[1]}]$  as described in Appendix A and the lower bound (10) using R (see Appendix B). The computing time depends on  $K$ , the total number of combinations generated. Table 4 shows the performance of solutions obtained with the exact moment only up to  $K = 16$  given the high computational times due to the multidimensional integrals involved. In contrast, it is notable how the computing time required for finding solutions using the moment

$H$ (max. # shut, $s$ )	$K$	Time using $LB(E[X_{[1]}])$ in (6)	Time using exact $E[X_{[1]}]$ in (6)	$\overline{\text{MSE}}$ ( $\widehat{\text{sd}}(\text{MSE})$ ) using exact $E[X_{[1]}]$	$\overline{\text{MSE}}$ ( $\widehat{\text{sd}}(\text{MSE})$ ) using $LB(E[X_{[1]}])$	P-value for $H_1$ : exact $E[X_{[1]}]$ better
4(2)	11	5.2	457.5	772.2 (7.32)	800.4 (5.78)	0.0000
4(3)	15	6.0	2001.3	701.3 (10.06)	672.6 (18.10)	1.0000
5(2)	16	6.1	3167.2	349.8 (1.54)	391.8 (2.28)	0.0000

Table 4: Computing times (in seconds) for solving the setup weigher problem (6)-(9) using the exact or lower bound for the smallest order statistic  $E[X_{[1]}]$  for problems with few combinations  $K$ , considering combinations where up to  $s$  hoppers remain shut in a cycle. In all cases, 100 initial trials  $T = 500$  and  $\alpha = 0.123$ . The average and standard errors of MSEs were estimated based on 50,000 simulated cycles. Times on an Intel 2.93 GHz Core 2 PC running R.

lower bound scales well with  $K$  (Table 5). For the three cases where we are able to compare the performance of the heuristic with the exact moment and with its lower bound approximation, the differences in MSE are inconclusive about which one is better overall. The exact moment provided an statistical better MSE solution in the cases  $H = 4$  (up to 2 shut) and  $H = 5$  (up to 2 shut), but the lower bound approximation provides a better MSE solution in the case  $H = 4$  (up to 3 shut).

As the number of hoppers increases and one considers combinations where more hoppers  $s$  remain shut, a numerical problem occurs:  $\Sigma$  and hence  $\mathbf{M}$  become very ill-conditioned, up to a point when  $\det(\Sigma)$  is numerically zero. In our experiments this occurred (for  $\alpha = 0.123$ ) when  $H > 10$  and all  $comb(H, 3)$  were considered. This ill-conditioning can be reduced if the combination that consists in all hoppers opening is not considered. This was implemented in our computer code (Appendix B), which permitted us to solve problems for  $H > 10$ .

## 4.2 Discussion: Recommendations for setting up an MWM

The characterization of an optimal solution given in section 3 provides useful insights for how to setup an MWM. In particular, the hopper weights should generate several different combinations of sums of weights whose densities cluster around the target package weight ( $T$ ) providing many feasible options to the built-in knapsack algorithm to choose from (characteristic # 2). In order to achieve this, the setting up of the hopper weights should form a very specific “ladder” of weights  $\{\mu_i\}$  whose values can be obtained with the heuristic in section 4 and the accompanying R program (see Appendix B).

In addition, there is always one hopper weight that should contain considerable less product, to help complete a package weight closer to the target a higher proportion of the times. Table 6 shows the solutions obtained in one of the replicated optimizations conducted for the cases in Tables 4 and 5. As it can be seen, the last hopper is set to a substantially lower target mean weight  $\mu_H$  than for the rest of the hoppers. We observed, via simulation, that if the solutions in Table 6 are implemented, the hopper with the minimum weight will *always* open in every cycle because it “completes” the package weight to the desired target.

$H$ (max, # shut, $s$ )	$K$ (# combs.)	Solution time	Simulated performance	
			$\overline{\text{MSE}}$	$\widehat{\text{sd}}(\text{MSE})$
6(2)	22	6.2	133.8	1.58
7(2)	29	7.7	102.4	0.80
8(2)	37	10.5	80.5	0.92
6(3)	42	10.7	100.4	0.88
9(2)	46	16.9	111.9	1.20
7(3)	64	13.9	34.3	0.68
12(2)	79	17.1	12.8	0.27
8(3)	94	20.9	23.5	1.24
15(2)	121	31.8	9.44	0.41
9(3)	130	36.7	8.98	0.34
12(3)	299	162.1	1.23	0.02
15(3)	575	355.2	0.63	0.03

Table 5: Computing times (in seconds) for obtaining a solution with the lower bound heuristic and the resulting average and estimated standard error of the MSE values over 10 simulations for larger values of the number of combinations ( $K$ ). Times and MSEs comparisons for solutions obtained solving (6)-(9) with the lower bound  $LB(E[X_{[1]}])$  approximation in (6). Other parameters were  $\alpha = 0.123$  and  $f = 0.3$  (in (8)) for all cases except 6(3), 7(3) and 9(2) where  $f = 0.4$  gave better solutions. MSEs estimated based on 50,000 simulated cycles (if  $K \leq 50$ ) or 10,000 cycles (if  $K \geq 50$ ). Times on an Intel 2.93 GHz Core 2 PC running R.

$H(s)$	Best weights found														
4(2)	271.3	255.8	237.8	65.6											
4(3)	470.7	450.3	80.3	48.3											
5(2)	242.0	183.8	160.8	125.5	90.5										
6(2)	149.9	149.2	133.9	127.6	108.2	69.7									
7(2)	146.0	141.0	106.2	101.6	94.5	81.5	77.2								
8(2)	140.1	102.4	101.8	86.5	79.8	77.4	74.2	56.0							
6(3)	199.6	186.7	171.3	148.3	136.4	52.7									
9(2)	143.9	119.7	108.3	102.4	98.4	46.6	33.9	24.1	19.4						
7(3)	173.8	154.4	142.7	129.9	120.7	89.2	60.0								
12(2)	106.7	87.6	82.3	66.3	53.5	48.8	47.8	43.0	34.1	31.9	30.5	8.0			
8(3)	128.0	126.7	113.8	110.6	107.0	103.9	89.8	65.2							
9(3)	146.6	119.6	105.2	93.1	84.2	75.3	69.6	67.9	4.5						
12(3)	113.0	105.6	95.8	76.7	67.0	62.8	51.4	37.2	36.7	27.6	26.3	9.1			
15(2)	100.1	79.5	70.0	64.9	60.4	55.5	45.4	34.4	31.6	29.3	18.4	15.7	11.1	8.6	6.5
15(3)	142.0	67.4	58.0	56.0	55.8	54.1	46.9	45.3	35.9	24.0	21.0	14.9	11.9	9.0	7.1

Table 6: Best solutions found with our lower bound heuristic corresponding to one of the ten replicates ran in the cases included in Tables 4 and 5.



1  
2  
3  
4 A notable result from our numerical experiments in Table 5 is how adding more com-  
5 binations by increasing the maximum number of hoppers shut (from  $s = 2$  to  $s = 3$  shut)  
6 but keeping the number of hoppers constant has a drastic decreasing effect in the average  
7 MSE, e.g., from 80.5 to 23.5 ( $H = 8$ ), from 111.9 to 8.98 ( $H = 9$ ), and from 12.8 to  
8 1.23 ( $H = 12$ ). *For most practical purposes, the MSE's obtained (one gram of product)*  
9 *in the types of products packed with MWMs (with no package underweighted) implies that*  
10 *the solutions obtained with the proposed heuristic method are "optimal enough".* These  
11 solutions (and the corresponding MSE analyses) can also be used to justify the acquisition  
12 of MWMs with large numbers of hoppers.  
13  
14  
15  
16

## 17 5 Conclusions and further research

18 In this paper, the setup problem of a multihead weigher machine has been studied. The  
19 hopper weight settings provided by the proposed optimization approaches will result in  
20 substantial savings over standard ad-hoc setup procedures used for companies utilizing  
21 MWMs.  
22  
23  
24  
25

26 A heuristic optimization model was developed based on a detailed characterization of  
27 what constitutes a near optimal solution to the MWM setup problem. The heuristic re-  
28 quires computations of moments of order statistics of correlated variables, and this becomes  
29 computationally intractable even for moderate size problems. Using a lower bound approx-  
30 imation of the moments of smallest order statistics proved to be considerably faster. The  
31 lower bound heuristic is applicable for MWMs with several hoppers, providing excellent  
32 solutions by considering only combinations that leave up to 3 hoppers shut in a cycle,  
33 reducing the search space considerably.  
34  
35

36 The behavior of the optimal solutions for MWM's with different number of hoppers  
37 indicate how the optimal setpoint weights per hopper decrease as the number of hoppers  
38 increases, with the weights in the best solutions found always forming a "ladder" of decreas-  
39 ing weights, and one hopper with considerably lowest weight. It was shown how the mean  
40 square error of the packed weights decreases as the number of combinations of hoppers  
41 increases. Furthermore, great accuracy in packaging with minimum product "given away"  
42 can be obtained with an optimally setup MWM with a large number of hoppers ( $H > 10$ )  
43 if only combinations with up to 3 hoppers remaining shut are considered. The analysis  
44 presented in this paper may also be used to justify the adoption of advanced MWMs with  
45 several hoppers currently available in the market. We make available an implementation  
46 of the proposed lower bound heuristic written in the R language (see Appendix B and  
47 supplementary material) which could be used for either tuning an MWM at startup or to  
48 justify the purchase of a machine with more hoppers.  
49  
50  
51  
52

53 Further research can be directed to study other types of MWMs with more complex  
54 architecture and to study optimal setup problems of MWMs under objective functions dif-  
55 ferent than the mean square error criterion investigated herein. For more complex MWMs  
56 with a very large number of hoppers  $H$ , or that mix several types of product in the same  
57 package, a simulation-optimization approach may be necessary to find its optimal settings.  
58 Beretta et al. [5] have recently developed a simulation-optimization approach for MWMs  
59 which includes economic considerations such as costs assigned to overfilled packages (rather  
60  
61  
62  
63  
64  
65

1  
2  
3  
4 than the MSE objective used here) and other operational costs, as commonly done in the  
5 canning literature (e.g., [2, 13]).  
6  
7

## 8 9 10 **References**

- 11  
12 [1] Afonja, B., (1972), “The Moments of the Maximum of Correlated Normal and t-  
13 Variates”, *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 34,  
14 No. 2, pp. 251-262.  
15  
16 [2] Arcelus, F.J., and Rahim, M.A., (1996), “Reducing performance variation in the can-  
17 ning problem”, *European Journal of Operational Research*, 94, pp. 477-487.  
18  
19 [3] Beretta, A. (2010). *Procedura RSM per la configurazione di una pesatrice multitesta*,  
20 Unpublished Master’s thesis, (in Italian), Facolta’ di Ingegneria dei Sistemi, Politec-  
21 nico di Milano, Milano, Italy.  
22  
23 [4] Beretta, A., and Semeraro, Q., (2012), “On a RSM Approach to the Multihead  
24 Weigher Configuration”, *Proceedings of the 11th Biennial Conference on Engineer-*  
25 *ing Systems Design and Analysis, ASME 2012*, Vol. 1, pp. 225-233.  
26  
27 [5] Beretta, A., Semeraro, Q., and Del Castillo, E., (2016), “On the Multihead Weigher  
28 Machine Setup Problem”, *Packaging Technology and Science*, 29, pp. 175-188.  
29  
30 [6] Bertsimas, D., Natarajan, K., and Chung-Piaw, T., (2006), “Tight Bounds on Ex-  
31 pected Order Statistics”, *Probability in the Engineering and Informational Sciences*,  
32 Vol. 20, No. 4, pp. 667-686.  
33  
34 [7] James, R.J.W., and Storer, R.H., (2005), “Techniques for solving subset sum prob-  
35 lems within a given tolerance,” *International Transactions in Operational Research*,12,  
36 pp.437-453.  
37  
38 [8] Kameoka, K., Nakatani, M., and Inui, N., (2000). “Phenomena in probability and  
39 statistics found in a combinatorial weigher” (in Japanese). *Transactions of the Society*  
40 *of Instrument and Control Engineers*, Vol. 36, pp. 388394.  
41  
42 [9] Karuno, Y., Nagamochi, H., and Wang, X., (2007). “Bi-criteria food packing by dy-  
43 namic programming”, *J. of the Operations Research Society of Japan*, 50(4), pp. 376-  
44 389.  
45  
46 [10] Karuno, Y., Nagamochi, H., and Wang, X., (2010). “Optimization problems and al-  
47 gorithms in double-layered food packing systems”, *J. of Advanced Mechanical Design,*  
48 *Systems, and Manufacturing*, 4(3), pp. 605-615.  
49  
50 [11] Imahori, S., Karuno, Y., Nagamochi, H., and Wang, X., (2011). “Kansei engineering,  
51 humans and computers: efficient dynamic programming algorithms for combinatorial  
52 food packing problems”, , *Int. J. of Biometrics*, 3(3), pp. 228-245.  
53  
54 [12] Ishida Corporation Ltd., Japan, <http://www.ishida.com/>  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

- 1  
2  
3  
4 [13] Pollock, S., and Golhar, D., (1998). “The canning problem revisited: the case of  
5 capacitated production and fixed demand”, *European Journal of Operational Research*,  
6 105, pp. 475-482.  
7  
8  
9 [14] Leppard, P., and Tallis, G.M., (1989), “Algorithm AS 249: Evaluation of the Mean  
10 and Covariance of the Truncated Multinormal Distribution”, *Journal of the Royal*  
11 *Statistical Society. Series C (Applied Statistics)*, Vol. 38, No. 3, pp. 543-553.  
12  
13 [15] Prekopa, A. (1995). *Stochastic Programming*, Kluwer Academic Publishers, Dordrecht,  
14 Boston.  
15  
16 [16] Tallis, G.M., (1965), “Plane Truncation in Normal Populations”, *Journal of the Royal*  
17 *Statistical Society. Series B (Methodological)*, Vol. 27, No. 2, pp. 301-307.  
18  
19 [17] Wilhelm, S., (2013), “Package tmvtnorm”, March 30, 2013,  
20 <http://www.r-project.org/> downloaded April 2013.  
21  
22 [18] Wang, B., and Mazumder, P., (2005), “Multivariate Normal Distribution Based Statis-  
23 tical Timing Analysis Using Global Projection and Local Expansion”, *Proceedings of*  
24 *the 18th International Conference on VLSI Design held jointly with 4th International*  
25 *Conference on Embedded Systems Design (VLSID05)*,  
26  
27  
28  
29  
30

## 31 Appendix A. Exact computations for the moments of 32 the minimum of $K$ unconstrained, correlated normals 33 34

35 Let  $\mathbf{X}_{[1]} = \min(X_1, X_2, \dots, X_K) = \min(\mathbf{X})$ . Theorem 1 below provides expressions for  
36  $E(X_{[1]})$  and  $\text{Var}(X_{[1]})$  when  $\mathbf{X}$  is a multivariate normal with arbitrary mean and covariance  
37 matrix, not subject to any constraint. Afonja [1] provided expressions for the computation  
38 of the moments of the maximum order statistics, which we modify in the Theorem below  
39 for the moments of the minimum order statistic. In the theorem,  $\phi_K(\mathbf{x}; \boldsymbol{\Theta}, \boldsymbol{\Sigma})$  denotes the  
40 (multivariate normal) pdf of  $\mathbf{X}$  with  $\boldsymbol{\Theta} = (\theta_1, \dots, \theta_K)'$ ,  $\boldsymbol{\Sigma} = \{\sigma_{ij}\}$ , and  $\phi_K(\mathbf{Z}; \mathbf{R})$  denotes  
41 the pdf of a standard multivariate normal with correlation matrix  $\mathbf{R}$ .  
42  
43  
44

45 **Theorem 1** . Let  $\mathbf{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  be a  $K$ -dimensional normal random variable. The  $r^{\text{th}}$   
46 moment of  $X_{[1]}$  about the origin is given by  
47  
48

$$49 \quad m'_r(X_{[1]}) = \sum_{i=1}^K \sum_{j=0}^r \binom{r}{j} \theta_i^{r-j} \sigma_i^j m_j(Z_i) \mathbf{b}_i \mathbf{R}_i$$

50 where  $m_j(Z_i)$  denotes the marginal  $j^{\text{th}}$  moment of a truncated standardized multivariate  
51 normal which is given by  
52  
53

$$54 \quad m_j(Z_i) \mathbf{b}_i \mathbf{R}_i = \int_{-\infty}^{b_{i1}} \int_{-\infty}^{b_{i2}} \cdots \int_{-\infty}^{b_{iK}} Z_i^j \phi_K(\mathbf{Z}; \mathbf{R}_i) d\mathbf{Z}$$

with the upper limits of integration  $\mathbf{b}'_i = (b_{i1}, \dots, b_{iK})$  equal to:

$$b_{ij} = \begin{cases} \frac{\theta_j - \theta_i}{\sqrt{\sigma_{ii} + \sigma_{jj} - 2\sigma_{ij}}}, & j \neq i \\ \infty, & j = i \end{cases} \quad (11)$$

and the correlation matrix  $\mathbf{R}_i$  is given by:

$$\mathbf{R}_i = \left\{ \frac{\sigma_{ii} - \sigma_{ij'} - \sigma_{ij} + \sigma_{jj'}}{\sqrt{\sigma_{ii} + \sigma_{jj} - 2\sigma_{ij}} \sqrt{\sigma_{ii} + \sigma_{j'j'} - 2\sigma_{ij'}}} \equiv r_{i,jj'} \right\}_{j' \neq i, j \neq i} \quad (12)$$

**Proof.** If  $X_{[1]} = \min(\mathbf{X})$  then

$$\mathbb{E}(X_{[1]}^r) \equiv m'_r(X_{[1]}) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \min(\mathbf{X})^r \phi_K(\mathbf{X}; \Theta, \Sigma) d\mathbf{X}$$

where  $X_{[1]} = X_i$  in the region  $A_i = \{\mathbf{X} : X_i < X_j; \forall j \neq i, -\infty < X_i < \infty\}$  with  $\bigcup_{i=1}^K A_i = \mathbb{E}^K$  and  $A_i \cap A_j = \emptyset (i \neq j)$ , except for sets of zero measure where  $X_i = X_j$  for some  $i, j$ . Note how regions  $A_i$  correspond, for  $K = 2$ , to areas number 1 and 4 in Figure 3 for the unconstrained case, i.e., when  $T = 0$ . Therefore,

$$\begin{aligned} \mathbb{E}(X_{[1]}^r) &= \int_{\bigcup_{i=1}^K A_i} \min(\mathbf{X})^r \phi_K(\mathbf{X}; \Theta, \Sigma) d\mathbf{X} \\ &= \int_{A_1} X_1^r \phi_K(\mathbf{X}; \Theta, \Sigma) d\mathbf{X} + \int_{A_2} X_2^r \phi_K(\mathbf{X}; \Theta, \Sigma) d\mathbf{X} + \cdots + \int_{A_K} X_K^r \phi_K(\mathbf{X}; \Theta, \Sigma) d\mathbf{X} \\ &= \sum_{i=1}^K \int_{A_i} X_i^r \phi_K(\mathbf{X}; \Theta, \Sigma) d\mathbf{X} \\ &= \sum_{i=1}^K \int_{A_i} (\theta_i + \sigma_i Z_i)^r \phi_K(\mathbf{Z}; \mathbf{R}) d\mathbf{Z} \end{aligned}$$

where the last equality follows from  $Z_i = (X_i - \theta_i)/\sigma_i$  ( $\sigma_i \equiv \sqrt{\sigma_{ii}}$ ),  $i = 1, \dots, K$ .

Following Wang and Mazumder (2005), we transform the integration region by redefining the  $Z_j$  variables for  $j \neq i$  according to

$$A_i = \left\{ \mathbf{Z} : -\infty < Z_j \equiv \frac{X_i - X_j - (\theta_i - \theta_j)}{\sqrt{\text{Var}(X_i - X_j)}} < \frac{\theta_j - \theta_i}{\sqrt{\text{Var}(X_i - X_j)}} \equiv b_{ij}; \forall j \neq i, -\infty < Z_i \equiv \frac{X_i - \theta_i}{\sigma_i} < \infty \right\}$$

and form the  $K \times 1$  vector

$$\mathbf{Z} = \begin{pmatrix} Z_i \equiv \frac{X_i - \theta_i}{\sigma_i} \\ \hline Z_1 \equiv \frac{X_i - X_1 - (\theta_i - \theta_1)}{\sqrt{\text{Var}(X_i - X_1)}} \\ Z_2 \equiv \frac{X_i - X_2 - (\theta_i - \theta_2)}{\sqrt{\text{Var}(X_i - X_2)}} \\ \vdots \\ Z_K \equiv \frac{X_i - X_K - (\theta_i - \theta_K)}{\sqrt{\text{Var}(X_i - X_K)}} \end{pmatrix} \quad (13)$$

where the 2nd to  $K$ th elements below the line do not include  $Z_i$ . Denote by  $\mathbf{R}_i$  the  $K \times K$  covariance matrix of  $\mathbf{Z}$  with entries as in (12), which, from the definition of  $\mathbf{Z}$  (13) equals to:

$$\mathbf{R}_i = \{\text{Corr}(X_i - X_j, X_i - X_j)\}_{j \neq i, j' \neq i}.$$

We then have:

$$E(X_{[1]}^r) = \sum_{i=1}^K \int_{-\infty}^{\mathbf{b}_i} (\theta_i + \sigma_i Z_i)^r \phi_K(\mathbf{Z}; \mathbf{R}_i) d\mathbf{Z}$$

where  $\mathbf{b}'_i = (b_{i1}, b_{i2}, \dots, b_{iK})$  ( $b_{ii} = \infty$ ) as in (11). Evaluating the binomial term inside the integral we finally get:

$$E(X_{[1]}^r) \equiv m'_r(X_{[1]}^r) = \sum_{i=1}^K \sum_{j=0}^r \binom{r}{j} \theta_i^{r-j} \sigma_i^j m_j(Z_i) \mathbf{b}_{\mathbf{R}_i} \quad (14)$$

where the moments of a truncated, standard multivariate normal are

$$m_j(Z_i) \mathbf{b}_{\mathbf{R}_i} = \int_{-\infty}^{\mathbf{b}_i} Z_i^j \phi_K(\mathbf{Z}; \mathbf{R}_i) d\mathbf{Z} \quad \text{QED.} \quad (15)$$

## Computational details for the first two moments of the smallest order statistic

Formula (14) in Theorem 1 gives for  $r = 1$ :

$$E(X_{[1]}) = E[\min(\mathbf{X})] = \sum_{i=1}^K \left( \theta_i m_0(Z_i) \mathbf{b}_{\mathbf{R}_i} + \sigma_i m_1(Z_i) \mathbf{b}_{\mathbf{R}_i} \right) \quad (16)$$

and for  $r = 2$ :

$$\begin{aligned} \text{Var}(X_{[1]}) &= \text{Var}[\min(\mathbf{X})] = E(X_{[1]}^2) - E(X_{[1]})^2 \\ &= \sum_{i=1}^K \left( \sigma_i^2 m_2(Z_i) \mathbf{b}_{\mathbf{R}_i} + 2\theta_i \sigma_i m_1(Z_i) \mathbf{b}_{\mathbf{R}_i} + \theta_i^2 m_0(Z_i) \mathbf{b}_{\mathbf{R}_i} \right) - E(X_{[1]})^2 \end{aligned} \quad (17)$$

Tallis [16, 14] provides algorithms for computing the moments of a truncated multivariate normal (implemented in R package `tmvtnorm`) which can in principle be used to compute (15) for  $j = 0, 1, 2$  and hence we would only compute (16) and (17) and be done, as suggested by Afonja [1]. However, this is too slow, and a better approach is to use a recursive result from Wang and Mazumder [18]. These authors perform the “trick” of putting the  $i$ th element of a vector in row 1 (as done in (13) above) which simplifies the handling of subindices in the computations required. The formulae by these authors have some typos that we correct below.

For any upper bounds  $\mathbf{B}$  and any standard normal random vector  $\mathbf{Y}$  with correlation matrix  $\mathbf{V}$ :

$$m_0(Y_1) \frac{\mathbf{B}}{\mathbf{V}} = \int_{-\infty}^{\infty} \underbrace{\int_{-\infty}^{b_1} \dots \int_{-\infty}^{b_j} \int_{-\infty}^{b_p}}_{j \neq i} \phi_p(\mathbf{Y}, \mathbf{V}) d\mathbf{Y} \quad (18)$$

(assuming hereon that all vectors have their element  $i$  put in element 1). Note that integration over  $i$  is for all reals, so this is really an integral over  $p - 1$  dimensions).

In addition, if  $\rho_{1,j} = \text{corr}(Y_1, Y_j)$  ( $j \neq i$ ) and  $\phi(b)$  is the standard normal density evaluated at  $b$ , Wang and Mazumder provide, for  $r \in \{1, 2\}$  the (here corrected) recursive expression:

$$m_r(Y_1) \frac{\mathbf{B}}{\mathbf{V}} = (r-1)m_{r-2}(Y_1) \frac{\mathbf{B}}{\mathbf{V}} + \sum_{j=2}^K \left[ \rho_{1,j} \phi(b_j) \sum_{l=0}^{r-1} \binom{r-1}{l} (1 - \rho_{1,j}^2)^{\frac{r-l-1}{2}} (-\rho_{1,j} b_j)^l m_{r-l-1}(Y_1) \frac{\mathbf{B}_j}{\mathbf{V}_j} \right] \quad (19)$$

where

$$\mathbf{B}_j = \left\{ \mathbf{Y}_{(j)} : -\infty < Y_1 < \infty, -\infty \leq Y_l \leq \frac{b_l - b_j \rho_{l,j}}{\sqrt{1 - \rho_{l,j}^2}}, l = 2, 3, \dots, K, l \neq j \right\}.$$

We define a vector  $\mathbf{Y}_{(j)}$ , consisting of all elements in  $\mathbf{Y}$  except of  $Y_j$ . Matrix  $\mathbf{V}_j$  in (19) contains the partial correlations between  $Y_m$  and  $Y_n$  given  $Y_j$ , for any  $Y_m$  and  $Y_n$  in  $\mathbf{Y}_{(j)}$  and is therefore of one dimension less than the covariance of the calling vector  $\mathbf{Y}$ :

$$\mathbf{V}_j = \left\{ \frac{\rho_{m,n} - \rho_{j,m} \rho_{j,n}}{\sqrt{1 - \rho_{j,m}^2} \sqrt{1 - \rho_{j,n}^2}} \right\}_{m,n \neq j} \quad (20)$$

where  $\rho_{m,n} \equiv r_{i,mn}$  computed as in (12) above.

Equation (19) results, for  $r = 1$  in:

$$m_1(Y_1) \frac{\mathbf{B}}{\mathbf{V}} = \sum_{l=2}^K \rho_{1,l} \phi(b_l) m_0(Y_1) \frac{\mathbf{B}_l}{\mathbf{V}_l} \quad (21)$$

and for  $r=2$  we get:

$$m_2(Y_1) \frac{\mathbf{B}}{\mathbf{V}} = m_0(Y_1) \frac{\mathbf{B}}{\mathbf{V}} + \sum_{j=2}^K \rho_{1,j} \phi(b_j) \left[ (1 - \rho_{1,j}^2)^{1/2} m_1(Y_1) \frac{\mathbf{B}_j}{\mathbf{V}_j} - (\rho_{1,j} b_j) m_0(Y_1) \frac{\mathbf{B}_j}{\mathbf{V}_j} \right] \quad (22)$$

Using (16), (18) and (21) we can obtain  $E(X_{[1]})$ . Substituting (21) and (22) into the variance formulae (17) we get  $\text{Var}(X_{[1]})$ . However, the computation of  $\text{Var}(X_{[1]})$  needs to be done with care as the  $m_r(Y)$  functions call each other recursively with vector and matrix arguments of decreasing dimension.

Specifically, in the second call to  $m_0(Y_1)$  we need to form the  $(K-2) \times 1$  vector  $\mathbf{Y}_{(j,l)}$ , a vector equal to  $\mathbf{Y}_{(j)}$  without element  $l$ .  $\mathbf{V}_{j,l}$  is then the  $(K-2) \times (K-2)$  partial correlation

matrix between the elements of  $\mathbf{Y}_{(j,l)}$  given  $Y_l$ . These partial correlations are obtained via (20) but using the entries in matrix  $\mathbf{V}_j$  rather than the entries in matrix  $\mathbf{R}_i$ . Likewise,  $\mathbf{B}_{j,l}$  is obtained using the expressions for the bounds in  $\mathbf{B}_j$ , but using the correlations in  $\mathbf{V}_j$  instead of those in  $\mathbf{R}_i$ .

We can now give a summary algorithm.

---

**Algorithm 1**  $E(X_{[1]})$  and  $\text{Var}(X_{[1]})$  computation

---

- 1: Given:  $\mathbf{X} \sim N(\boldsymbol{\Theta}, \boldsymbol{\Sigma})$ , a  $K$ -dimensional vector with  $\boldsymbol{\Theta}$  and  $\boldsymbol{\Sigma}$  known.
  - 2: **for**  $i = 1$  to  $K$  **do**
  - 3:      $\mathbf{Y} = \mathbf{Z}_i$ ,  $\mathbf{B} = \mathbf{b}_i$  using (11),  $\mathbf{V} = \mathbf{R}_i$  using (12). This requires putting element  $i$  in position 1, so  $Z_1 \leftarrow Z_i$ . This must be reflected in matrix  $\mathbf{R}_i$ .
  - 4:     Evaluate  $m_0(Z_1) \frac{\mathbf{B}}{\mathbf{V}}$  using (18)
  - 5:     Evaluate  $m_1(Z_1) \frac{\mathbf{B}}{\mathbf{V}}$  using (21). This turns requires  $m_0(Z_1) \frac{\mathbf{B}_j}{\mathbf{V}_j}$
  - 6:     Evaluate  $m_2(Z_1) \frac{\mathbf{B}}{\mathbf{V}}$  using (22). This turns requires  $m_1(Z_1) \frac{\mathbf{B}_j}{\mathbf{V}_j}$  which in turn requires  $m_0(Z_1) \frac{\mathbf{B}_{j,l}}{\mathbf{V}_{j,l}}$
  - 7:     Accumulate sums over  $i$  for  $E(X_{[1]})$  (16) and  $\text{Var}(X_{[1]})$  (17)
  - 8: **end for**
  - 9: Return (16) and (17), the first two moments of the smallest order statistics of the possibly correlated normal variables in  $\mathbf{X}$ .
- 

## Appendix B. R software implementation of heuristic method

The heuristic in section 4 was implemented in R (program `OptimizeWeigher.R`). This program contains function `constraintsAll` which evaluates the constraints (6)-(9) for given  $\boldsymbol{\mu}, T$ , and  $\alpha$ . Function `computeDet` evaluates the objective function (5). The program also allows the user to apply the modified lower bound heuristic which uses (10) in constraint (6) –using the heuristic is actually the default. The program contains function `computeMSD` (not used in the heuristic of section 4) which evaluates expressions for  $E[X_{[1]}]$ ,  $\text{Var}[X_{[1]}]$ ,  $E[X_{[K]}]$  and  $\text{Var}[X_{[K]}]$ , the first two moments of the two extreme order statistics of a general  $K$ -dimensional normal distributed variable, a function that may be useful in other Applied Statistical problems.