# Equal-Sized Partition Problem: Application in Spinning and Yarn Production

**Quoc-Trung Bui**
SoICT, Hanoi University of Science and Technology, Vietnam

**Abstract**

Equal-Sized Partition (ESP) problem originates from real-world yarn production planning, where cotton bales that are characterized by type, color, and micronaire must be selected and divided into equal-sized groups (one per production day) such that the counts of each type and color grade differ by at most 1 across groups, while minimizing the maximum difference in average Mic between any two groups. A spinning company in Hue province, Vietnam, previously relied on a manual three-step process to solve this challenge. This study proves the NP-hardness of ESP, introduces intelligent computational approaches for its complex steps, proposes a dynamic programming algorithm for exact micronaire discretization in the first step, and develops a constraint-based local search metaheuristic for assigning bales to days in the third step. Experiments on five months of historical company data show substantial improvements over the manual method, resulting in the adoption of the proposed solution for daily operations.

**2012 ACM Subject Classification** Theory of computation → Randomized local search; Applied computing; Mathematics of computing → Combinatorial optimization

**Keywords and phrases** Spinning and Yarn Production, Cotton Bale Allocation, NP-Hard, Constraint-Based Local Search, Dynamic Programming

**Digital Object Identifier** 10.4230/LIPIcs.CP.2026.9

**Funding** This research was funded by the Ministry of Education and Training of Vietnam under project code CT2025.EA.BKA.13.

## 1 Introduction

In this paper, we study the Equal-Sized Partition (ESP) problem, which arises in spinning and yarn production. Given a set $I = \{1, \ldots, n\}$ of $n$ cotton bales, the task is to **select** and simultaneously **partition** them into $m$ equal-sized subsets $\{S_1, \ldots, S_m\}$, where each subset $S_i$ contains exactly $p$ bales (with $n \geq m \cdot p$). Each bale $i \in I$ is characterized by a triple $\langle t_i, c_i, m_i \rangle$, where:

- $t_i \in T$ is the cotton type (from a predefined finite set $T$),
- $c_i \in C$ is the color grade (from a predefined finite set $C$),
- $m_i \in \mathbb{R}$ is the Micronaire [1] (abbreviated as Mic).

The partition must satisfy the following *balance constraints*: for every cotton type $t \in T$ and every color grade $c \in C$, the number of bales with type $t$ (respectively grade $c$) differs by at most 1 across the $m$ subsets $S_1, \ldots, S_m$. The objective is to minimize the range of the subset means of the Mic values. More precisely, we seek to minimize the maximum difference between any two subset averages:

[1] Micronaire measures the air permeability of a fixed mass of cotton (typically 10 g) compressed into a fixed volume. Air is forced through the sample, and a higher value indicates thicker/less fine cotton fibers (fewer air voids). It is one of the most important indicators of cotton quality, reflecting both fiber fineness (linear density) and maturity.


32nd International Conference on Principles and Practice of Constraint Programming (CP 2026).
Editor: Nicolas Beldiceanu; Article No. 9; pp. 9:1–9:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

$$\text{minimize} \quad \max_{i\in\{1,\dots,m\}} \left( \frac{1}{p} \sum_{j\in S_i} m_j \right) - \min_{i\in\{1,\dots,m\}} \left( \frac{1}{p} \sum_{j\in S_i} m_j \right)$$

The ESP problem belongs to the classes of subset selection problems and partition problems [8, 10] and can be viewed as a generalization of the well-known 3-Partition problem [4, 5], $k$-partitioning problem [1], which are strongly NP-complete [5]. Consequently, ESP is NP-hard. A formal proof of NP-hardness is provided in Subsection A.1 of the Appendix.

In the literature, there exist problems that are similar to the ESP problem, such as the blending problem [7, 9, 2, 6, 3]. However, to the best of our knowledge, the ESP problem is a totally new one.

## 1.1 Real-Life Application in Spinning and Yarn Production

We collected and analyzed the business requirements from a spinning joint-stock company (listed on the UPCoM exchange) located in the Phu Bai Industrial Zone, Phu Bai Ward, Huong Thuy Town, Thua Thien Hue Province, Vietnam. The company's core business involves the manufacturing and export of various types of ring-spun yarn, such as 100% cotton yarn, primarily using raw cotton as the main input material.

The selling price of yarn depends heavily on its quality, which is largely determined by the uniformity of key raw cotton attributes, in particular, cotton type, color grade, and Mic. Yarn produced from bales exhibiting uniform levels of cotton color grade and Mic commands a significantly higher price, often several times higher, than yarn produced from bales with varying levels of these attributes. This price difference arises because inconsistent fiber quality leads to uneven dyeing (color patchiness or bleeding) during subsequent processing steps. Consequently, high-uniformity yarn is typically used in luxury fashion products, whereas lower-uniformity yarn is generally suitable only for low-value items such as towels.

The company regularly purchases raw *cotton bales* (approximately 200 kg each) of different types, color and Mic from multiple countries, such as India, Togo, and Mexico, for yarn production and sales. Therefore, preparing raw cotton bales for blending to ensure consistent material quality for daily yarn production is of critical importance. At the company under study, workers prepare one *cotton mixing table* (also called a cotton laydown) for each production day, consisting of 54 raw cotton bales arranged side by side on a designated table. The cotton-feeding head of the blending machine then traverses back and forth across the table, suctioning and mixing cotton from the bales. Figure 1 shows a photograph of cotton bales, a mixing table, and the cotton-feeding head of the blending machine at the company.

The assignment of specific bales to each production day is determined by the planning department. Upon arrival of a new cotton lot (typically sufficient for approximately one month of production), the planning department creates a bale allocation schedule for the corresponding production days. The main objective of this allocation is to achieve the highest possible uniformity in the daily blended cotton quality. The **current manual solution approach** for allocating bales to production days consists of three consecutive steps:

**First step.** Bales that come from the same farm, share the same grades of neps, trash content, and unevenness, and belong to the same *cotton type* are grouped together. The first step consists of classifying the cotton bales of each type into predefined *bale groups*. There are 12 predefined groups in total, formed as the cross-product of two cotton attributes: color grade (three levels: Dark, Light, Uniform) and Mic bin (four levels). Mic bins: Bin 1 ($\leq 3.5$), Bin 2 (3.51–4.0), Bin 3 (4.01–4.5), Bin 4 ($> 4.5$). ***The choice of 12 groups is justified by the need to balance operational simplicity with yarn quality assurance***.


**Figure 1** Photograph of the cotton bale, mixing table, and cotton-feeding head of the blending machine at the company under study.

While the use of three color grade levels is justified by the fact that most commercial cottons fall mainly into the "Uniform" color grade, the choice of four Mic bins is justified by the observation that Mic values for most commercial cottons typically range from 3.0 to 5.0. Table 1 illustrates the classification results for 252 bales imported from America, belonging to a lot of 1,250 bales.

**Second step.** Calculation of (i) the number of bales from each cotton type to be included in one mixing table, and (ii) the number of bales contributed by each bale group. These calculations are straightforward. The number of bales from a given type in one mixing table is the integer closest to $\frac{54 \cdot X}{Y}$, where $X$ is the number of bales of that type and $Y$ is the total number of bales across all types. For example, with 252 bales of America type in a lot of 1,250 bales, approximately 10 bales of America type are allocated per mixing table ($\lfloor 252 \times 54/1250 \rfloor = 10$). Next, the number of bales per bale group in the mixing table is calculated as the product of the type's bale count in the mixing table and the group's proportion within that type. Table 1 shows the resulting bale counts per group for the American type contributing 10 bales per mixing table (figures in parentheses).

**Third step.** Determination of the number of bales from each bale group of each cotton type assigned to each production day, based on the results from the previous step. In this step, cotton bales are used in their entirety and are not torn apart or divided. As the result of the previous step, if a bale group contributes $x$ bales per mixing table, $x$ consists of an integer part and a fractional part. The integer part is allocated consistently every day. The fractional part, however, requires random allocation. For example, if the fractional part is 0.21, one additional bale from that group is included approximately $1/0.21 \approx 5$ days.

Unfortunately, all the above planning activities are currently performed *manually with the support of Microsoft Excel*. As a result, determining the optimal set of bales for each production day remains a challenging and time-consuming task. The limitations of this process will be presented in the next section.

## 1.2 Objective and Contributions

The objective of this study is to improve the current manual solution approach that is currently being carried out at the spinning company, in the hope of increasing the quality of its produced yarn. The contributions of this paper include (1) introducing the ESP

**Table 1** Distribution of 252 American cotton bales (part of a larger lot of 1,250 bales) across the 12 bale groups, and the expected number of bales per group when American cotton contributes 10 bales per mixing table.

| | Dark grade | Light grade | Uniform grade |
|---|---|---|---|
| **Mic bin 1** | 1 (0.04) | 0 (0) | 4 (0.16) |
| **Mic bin 2** | 0 (0) | 5 (0.20) | 42 (1.67) |
| **Mic bin 3** | 10 (0.40) | 20 (0.79) | 68 (2.70) |
| **Mic bin 4** | 2 (0.08) | 30 (1.19) | 70 (2.78) |

problem, with its significant real-life application, to the literature, (2) proving its NP-hard complexity, (3) improving the current manual solution approach by defining two optimization subproblems, (4) proposing a dynamic programming algorithm and a constraint-based local search for the subproblems, and (5) conducting experiments on real-world instances to demonstrate the effectiveness of the proposed solution approach.

# 2 Proposed Solution Approach

## 2.1 Limitations of the Current Manual Solution Approach

The current solution approach presented in previous section has the following limitations:

- The discretization of Mic values into four fixed bins in the first step does not take into account the actual distribution of Mic values within the lot. Moreover, while bins 2 and 3 are fixed at a width of 0.5 (corresponding to the threshold expected by the company's board of directors), bins 1 and 4 can be significantly wider when many bales have Mic values well below 3.0 or well above 5.0. For example, among the 7,557 bales from five lots used in the five instances described in detail in Section 5, 272 bales have Mic values either below 3.0 or above 5.0. These lead to large variance in Mic values within the same bin, which directly reduces yarn quality.
- The allocation of the number of bales from each bale group of each cotton type in Step 3 is performed manually, with only Microsoft Excel as support. This practice entails many operational risks. In particular, for a given cotton type, the number of bales assigned to mixing tables across different production days can vary significantly (differences larger than 1 are frequently observed). The same issue occurs with respect to cotton color. Experimental results in Section 5, together with observations from actual operations, confirm that differences of two are common.
- Both the fixed Mic binning and the manual bale assignment for production days result in considerable variance in the average Mic values across mixing tables.

## 2.2 New Solution Approach

Following in-depth discussions with the company's board of directors and the staff responsible for mixing table schedule activities, and with the aim of avoiding major disruption to the daily work routines of the operational staff, the proposed solution retains the existing three-step process. However, intelligent computational methods are introduced in these steps to significantly improve operational quality. Among the three steps, Step 2 is straightforward and has no decisive impact on operational quality.

For Step 1 of the current manual solution, we address the limitation of fixed bins. Specifically, bins are now determined dynamically based on the actual Mic distribution, with the objective of minimizing the total standard deviation within all bins in hope of minimizing the total variance of mixing table Mic values. This approach received strong support from all operational staff and from the board of directors. This problem is referred to as the **Constrained Mic Binning** (CMB) problem and is defined as follows:

▶ **Definition 1** (CMB). *Given a non-decreasing sequence $M = \langle m_1, m_2, \ldots, m_n\rangle$ of $n$ Mic values, the task is to partition the sequence into at most $b^{\max}$ contiguous bins such that every bin satisfies the maximum difference constraint ($\max - \min \leq \Delta_{\mathrm{mic}}^{\max}$ within each bin), and the sum of the standard deviations of Mic values across all bins is minimized.*

For Step 3 of the current manual solution approach, when a bale group of a given cotton type contributes $x$ bales per mixing table (where $x$ consists of an integer part and a fractional part), the integer part is allocated uniformly, while the fractional part requires random assignment. The proposed method focuses on intelligently allocating this fractional part. This approach also received strong support from all operational staff and from the board of directors. This allocation task is denoted as the **Bale Allocation for Production Days** (BAPD) problem and is defined as follows:

▶ **Definition 2** (BAPD). *Given a set of $n$ classes of cotton bales distinguished by cotton type, cotton color grade, and bale group (each class containing multiple physical bales), determine how to select and assign physical bales to the remaining slots $\alpha$ of $m$ mixing tables (one table per production day) such that the number of physical bales of each cotton type and each cotton color differs by at most one across mixing tables, and the maximum difference in average Mic value among mixing tables is minimized.*

Note that, in general, the BAPD problem is NP-hard, as it reduces to the ESP problem in the special case where each class contains exactly one physical bale. These two problems will be addressed independently in the subsequent sections of this study.

# 3 Dynamic Programming for the CMB Problem

We propose, in this section, a Dynamic Programming Algorithm (DPA) to address the CMB problem to optimality.

### Subproblem Definition

For $1 \leq i \leq n$ and $1 \leq k \leq b^{\max}$, let $SP(i,k)$ be the following subproblem: Divide the prefix $\langle m_1, m_2, \ldots, m_i\rangle$ into *exactly* $k$ non-empty contiguous bins such that

- each bin satisfies the maximum difference constraint ($\max - \min \leq \Delta_{\mathrm{mic}}^{\max}$),
- the sum of the standard deviations over the $k$ bins is minimized.

Let $DP[i][k]$ denote the minimum total standard deviation achievable for $SP(i,k)$.

### Dynamic Programming Recurrence

**Base cases.**

- $DP[i][0] = +\infty$ for all $i \geq 1$ (impossible to cover with 0 bins)
- $DP[0][0] = 0$
- $DP[0][k] = +\infty$ for $k \geq 1$
- $DP[i][1] = \begin{cases} \sqrt{\frac{1}{i}\sum_{p=1}^{i}\left(m_p - \frac{\sum_{p=1}^{i} m_p}{i}\right)^2} & \text{if } m_i - m_1 \leq \Delta_{mic}^{max} \\ +\infty & \text{otherwise} \end{cases}$

**Recurrence (for $k \geq 2$).**

$$DP[j][k] = \min_{\substack{i=k-1 \\ m_j - m_{i+1} \leq \Delta_{\text{mic}}^{\max}}}^{j-1} \left( DP[i][k-1] + \text{stddev}(m_{i+1}, \ldots, m_j) \right)$$

where

$$\text{stddev}(m_l, \ldots, m_r) = \sqrt{\frac{1}{r-l+1} \sum_{p=l}^{r} \left( m_p - \bar{m}_{[l..r]} \right)^2}$$

and $\bar{m}_{[l..r]} = \frac{\sum_{p=l}^{r} m_p}{r-l+1}$ or the mean of $m_l, \ldots, m_r$.

There are $O(n \cdot b^{\max})$ states. For each state $DP[j][k]$, we consider up to $O(n)$ possible split points $i$. An implementation therefore yields a time complexity of $O(n^2 \cdot b^{\max})$.

# 4 Constraint-Based Local Search for the BAPD Problem

We have proposed a mixed integer programming formulation for the BAPD problem and implemented it using the CPLEX solver (Subsection A.2 of the Appendix). Unfortunately, it could not solve the real-life instances. Therefore, in this study, we propose a constraint-based local search (CBLS) approach to solve large-scale real-life instances of the problem.

## 4.1 Solution Encoding and Neighborhood Definition

A solution to the BAPD problem is represented as a set

$S = \left\{ \{b_1^1, \ldots, b_\alpha^1\}, \ldots, \{b_1^m, \ldots, b_\alpha^m\} \right\}$. In this encoding:

- $b_i^k$ ($1 \leq k \leq m$, $1 \leq i \leq \alpha_k$) denotes a cotton bale belonging to class, where each class corresponds to a unique combination of cotton type, color grade, and Mic bin.
- each subset (mixing table / production day) contains exactly $\alpha$ distinct cotton bales;
- $S$ contains exactly $m$ such subsets, corresponding to the $m$ mixing tables.

The constraint-based local search (CBLS) considers two main neighborhoods:

- **Substitution-based neighborhood** ($N_1$): A neighbor is obtained by replacing one cotton bale in one mixing table with another available bale from a *different* class.

$$N_1(S) = \left\{ S' \;\middle|\; S' \text{ differs from } S \text{ by substituting } b_i^k \text{ with } c_i^k \ (c_i^k \neq b_i^k, \text{or} \right.$$
$$\left. c_i^k \text{ belongs to a different class than } b_i^k) \right\}$$

- **Swap-based neighborhood** ($N_2$): A neighbor is obtained by swapping two bales from two different mixing tables, where the two bales also belong to different classes.

$$N_2(S) = \left\{ S' \;\middle|\; S' \text{ is obtained by swapping } b_i^p,\ b_j^q\ (p \neq q),\ b_i^p,\ b_j^q \text{ belong to different classes} \right\}$$

## 4.2 Two-Stage Local Search

The search in the CBLS method consists of two consecutive stages with distinct purposes. The first stage has the responsibility of finding a feasible solution that satisfies the constraints (the number of physical bales of each cotton type and each color grade differs by at most one across mixing tables); while the second stage improves the quality (objective value) of feasible solutions while maintaining feasibility. The detailed pseudo-code for all stages is given in Subsection A.3 of the Appendix.

**Initial solution generation.** The initial solution is generated in a greedy search. First, the maximum permissible number of bales of each type and each color grade per mixing table is precomputed as the ceiling of the total available bales divided by the number of mixing tables ($m$). The sets of cotton types and color grades are randomly shuffled to introduce stochasticity in the assignment order. Then, bales are processed group by group according to this random type–color sequence. For each group, the available bales are assigned one by one: at every step, a feasible mixing table is selected that still respects the type-capacity limit, the color-capacity limit, and the slot limit ($\alpha$), with the table chosen according to a lexicographic greedy criterion that first minimizes the current number of bales of the same type, then the number of bales of the same color, and finally the total number of occupied slots. This process continues until either all bales have been assigned or every mixing table has reached its full capacity of $\alpha$ slots, yielding a diversified yet constraint-compliant starting solution for the subsequent iterations.

**Stage 1 – Finding feasible solution.** This is a fast tabu-search repair procedure that converts the random initial solution $S$ into a fully feasible one. While any constraint violation remains, the algorithm evaluates the best substitution move in one non-tabu mixing table ($N_1$) and the best swap move between two non-tabu tables ($N_2$), then accepts the move yielding the smallest violation (preferring substitution on ties) and adds the involved table(s) to the tabu list. A stability counter tracks progress; improvements reset it, while stagnation increments it and triggers adaptive tabu-tenure updates. When the stability limit is reached or at periodic restart intervals, the search restarts from the best-known or a perturbed feasible solution. This guided mechanism quickly eliminates all violations and delivers a high-quality feasible starting point for subsequent improvement stages.

**Stage 2 – Improving obtained feasible solution.** This is a focused tabu-search improvement phase that takes a feasible solution and iteratively refines it to minimize the gap between the highest and lowest average MIC values across all mixing tables. In every iteration, the algorithm first identifies the table with the current highest average MIC ($t^+$) and the table with the lowest average MIC ($t^-$). It then evaluates three candidate moves: the best feasible substitution inside $t^+$ that most reduces the max–min MIC gap, the best substitution inside $t^-$, and the best swap between $t^+$ and $t^-$. The move with the largest positive improvement is selected according to a clear priority rule (substitution in $t^+$ is preferred, followed by substitution in $t^-$, then swap), the chosen move is performed, and the affected table(s) are added to the tabu list to avoid immediate cycling. The best solution found so far is updated whenever an improvement occurs, and a stability counter tracks consecutive non-improving iterations. If no progress is made, the tabu tenure is adaptively adjusted; when the stability limit is reached, the search restarts from the best-known solution or applies a strong perturbation. Additionally, every $restartFreq$ iterations the algorithm performs a full restart from a new random or lightly perturbed feasible solution, clears the tabu list, and resets the tenure to its middle value. This targeted intensification strategy efficiently balances the Mic distribution across tables while strictly preserving feasibility, producing higher-quality solutions within a limited number of iterations.

## 5 Experiments

The main goal of experiments is answer the following research question: *does the proposed solution approach reduce the constraint violation level and the variance in the Mic values of the mixing tables?* Five real-life instances were constructed by collecting historical data over a five-month period from the spinning company. A summary of these instances is provided in Table 2. Both the current manual solution approach and the newly proposed solution approach were re-implemented in C++ for evaluation and comparison purposes.

**Table 2** Summary of the five real-life instances. Columns **#B** and **#T** indicate the total number of cotton bales and the number of mixing tables, respectively. Columns **T1**–**T9** report the number of bales of each cotton type present in the instance (note that column labels **T1**–**T9** do not correspond to the same type across different instances).

| Inst. | # B | # T | Color | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 1750 | 32 | *Dark* | | | 9 | | | | | | |
| | | | *Light* | | | 153 | | | | | | |
| | | | *Uniform* | 120 | 226 | | 367 | 363 | 89 | 146 | 100 | 177 |
| **2** | 1309 | 24 | *Dark* | | | | | 111 | | | | |
| | | | *Light* | | | | | 136 | | | | |
| | | | *Uniform* | 120 | 249 | 88 | 243 | | 187 | 175 | | |
| **3** | 1455 | 26 | *Uniform* | 242 | 240 | 117 | 246 | 245 | 243 | 122 | | |
| **4** | 1250 | 23 | *Uniform* | 362 | 244 | 244 | 148 | 252 | | | | |
| **5** | 1712 | 31 | *Uniform* | 243 | 371 | 257 | 206 | 220 | 183 | 92 | 140 | |

For each instance, we ran the existing solution approach 10,000 times. For the proposed solution approach, we used the output from a single run of DPA and CBLS, then simulated the bale assignment to mixing tables 10,000 times. The main results are as follows:

- All solutions produced by the newly proposed approach satisfy all constraints. By contrast, all solutions generated by the current manual approach violate the constraint on the maximum allowable difference in the number of bales per cotton color grade for instances 1 and 2. For the remaining three instances, the manual solutions do not violate any constraints, as they involve only a single color grade.
- Histograms showing the distribution of the maximum difference between the average Mic values of bales assigned to any two mixing tables, across the 10,000 solutions for each instance, are presented in Figure 2. Further details, including the averages and standard deviations of these maximum differences, are reported in Table 3.

This experiment demonstrates that the proposed solution approach clearly outperforms the current manual approach. The solutions it produces exhibit significantly smaller maximum differences in average Mic values between any two mixing tables.

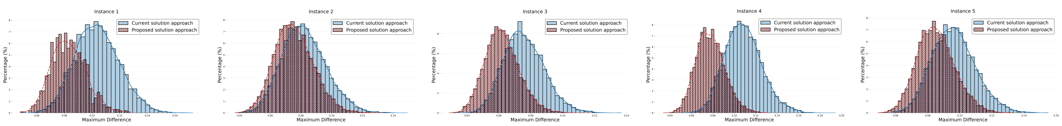

**Figure 2** Histograms of the maximum difference between the average Mic values of the bales assigned to any two mixing tables for 10,000 solutions for every instances.

## 6 Conclusion

In this paper, we addressed the Equal-Sized Partition (ESP) problem arising in the spinning and yarn production industry. We developed dynamic programming and constraint-based local search methods for its key subproblems in the current manual process employed at

**Table 3** Averages and standard deviations of maximum differences between the average Mic values of the bales assigned to any two mixing tables for 10,000 solutions for every instances.

| Instance | Current Solution Approach | | Proposed Solution Approach | |
|---|---|---|---|---|
| | *Avg. Max Diff* | *Std. Max Diff* | *Avg. Max Diff* | *Std. Max Diff* |
| 1 | 0.1037 | 0.0159 | 0.0830 | 0.0122 |
| 2 | 0.0821 | 0.0142 | 0.0744 | 0.0125 |
| 3 | 0.0759 | 0.0126 | 0.062 | 0.0106 |
| 4 | 0.1083 | 0.0171 | 0.0780 | 0.0132 |
| 5 | 0.0962 | 0.0134 | 0.0851 | 0.0116 |

a spinning mill in Hue province, Vietnam. The ESP problem was proven to be NP-hard. While one subproblem remains tractable via an exact method, the other is computationally intractable. Experimental evaluation on five real-world instances derived from five months of historical operational data demonstrated that the proposed computational methods produce significantly better solutions than the existing manual approach, particularly in terms of balancing cotton attributes and minimizing Mic variation across production groups.

For future work, we plan to extend the study to address emerging business requirements, such as optimized cotton bale stocking and inventory management strategies to further enhance the consistency and quality of the produced yarn.

## References


1. Luitpold Babel, Hans Kellerer, and Vladimir Kotov. Thek-partitioning problem. *Mathematical Methods of Operations Research*, 47(1):59–82, February 1998. doi:10.1007/BF01193837.
2. Victor CB Camargo, Bernardo Almada-Lobo, and Franklina MB Toledo. Integrated lot-sizing, scheduling and blending decisions in the spinning industry. *Pesquisa Operacional*, 41(spe):e233486, 2021.
3. Subhasis Das and Anindya Ghosh. Technique based on fuzzy logic for cotton bale lay-down management. *Fibres & Textiles in Eastern Europe*, 25(2 (122)), 2017.
4. Mauro Dell'Amico and Silvano Martello. Reduction of the three-partition problem. *Journal of Combinatorial Optimization*, 3(1):17–30, July 1999. doi:10.1023/A:1009856820553.
5. Michael R Garey and David S Johnson. *Computers and intractability*, volume 29. wh freeman New York, 2002.
6. Anindya Ghosh, Abhijit Majumdar, and Subhasis Das. A technique of cotton bale laydown using clustering algorithm. *Fibers and Polymers*, 13(6):809–813, 2012.
7. JH Greene, K Chatto, CR Hicks, CB Cox, TE Baynham, and TEJ Baynam. Linear programming-cotton blending and production allocation. *The Journal of Industrial Engineering*, 12:17, 1965.
8. Yaniv Hamo and Shaul Markovitch. The compset algorithm for subset selection. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, pages 728–733, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc. URL: http://ijcai.org/Proceedings/05/Papers/1574.pdf.
9. Huosong Xia, Yuan Wang, Sajjad Jasimuddin, Justin Zuopeng Zhang, and Andrew Thomas. A big-data-driven matching model based on deep reinforcement learning for cotton blending. *International Journal of Production Research*, 61(22):7573–7591, 2023. doi:10.1080/00207543.2022.2153942.
10. A. Çivril. Column subset selection problem is ug-hard. *Journal of Computer and System Sciences*, 80(4):849–859, 2014. doi:10.1016/j.jcss.2014.01.004.

# A Appendix

## A.1 NP-Hardness Analysis

Let ESP-$\Phi$ be the decision problem that asks whether there exists a solution to the ESP problem with objective value at most $\Phi$. We show that ESP-$\Phi$ is NP-complete by a polynomial-time reduction from the 3-Partition problem [5].

▶ **Theorem 3.** *The ESP-$\Phi$ problem is NP-complete.*

**Proof.** In the 3-Partition problem, we are given a set $S = \{1, \dots, n\}$ of $n$ elements, each element $i \in I$ is associated with a positive integral value of $a_i$ and an integer $m$ with $n = 3 \cdot m$. The task is to decide whether $S$ can be partitioned into $m$ disjoint subsets $S_1, \dots, S_m$ s.t.

$$|S_i| = 3 \quad \text{and} \quad \sum_{j \in S_i} a_j = B \quad \text{for all } i = 1, \dots, m,$$

where $B = \frac{1}{m} \cdot \left( \sum_{j=1}^{n} a_j \right)$ (i.e., all subsets have the same sum). We construct the following instance of ESP-$\Phi$ from the 3-Partition instance:

- There are $n$ cotton bales, indexed by $i = 1, \dots, n$;
- All bales have the same cotton type and color grade: $t_1 = \dots = t_n = \bar{t}$ and $c_1 = \dots = c_n = \bar{c}$;
- The Mic value $m_i$ of bale $i$ is $a_i$ for each $i$;
- The number of subsets is $m$; and
- The threshold is set to $\Phi = 0$.

($\Rightarrow$) Suppose the 3-Partition instance has a YES answer, i.e., there exists a partition of $\{1, \dots, n\}$ into $m$ triples $S_1, \dots, S_m$ with $\sum_{j \in S_i} a_j = B$ for all $i \in \{1, \dots, m\}$. Construct a solution for the ESP instance by assigning the three items corresponding to the numbers in $S_i$ to subset $i$ (for each $i = 1, \dots, m$). Then each subset $i$ has total Mic value $\sum m_j = B$. Thus the objective value is $0 \leq \Phi$, and the ESP-$\Phi$ instance is a YES instance.

($\Leftarrow$) Suppose the ESP-$\Phi$ instance is a YES instance, i.e., there exists a partition of the $n$ cotton bales into $m$ subsets of three elements $G_1, \dots, G_m$ such that the objective value is at most 0 (and therefore exactly 0). This implies that every subset $G_i$ has the same total Mic value $\sum_{j \in G_i} m_j = B$. Moreover, because the reduction sets $t_i = \bar{t}$ and $c_i = \bar{c}$ for all bales, the objective value is zero. Construct a 3-Partition solution by taking $S_i = \{j \mid \text{item } j \text{ is assigned to group } G_i\}$ for each $i$. Then each $S_i$ contains exactly three elements (since $n = 3 \cdot m$) and $\sum_{j \in S_i} a_j = B$. Thus the 3-Partition instance is also a YES instance.

Therefore, the 3-Partition instance is YES if and only if the constructed ESP-$\Phi$ instance is YES. Since the reduction is polynomial-time computable, hence ESP-$\Phi$ is NP-complete. ◀

## A.2 Mixed Integer Programming Formulation for the BAPD Problem

The mixed-integer programming formulation for the BAPD problem, introduced in Section 4, is presented in this subsection. It consists of four main components, as follows:

### A.2.1 Inputs

- A set $B$ of $n$ *classes of cotton bales*, where each class $b \in B$ is associated with:
  - a cotton type $b_t \in T$, where $T$ is a predefined finite set of cotton types,
  - a cotton color grade $b_c \in C$, where $C$ is a predefined finite set of color grades,

- a numerical Mic value $b_m \in \mathbb{R}^+$ that is the average Mic of the bin to which bale $b$ belongs,
- a positive integer $b_f \in \mathbb{Z}^+$ that is the number of physical cotton bales available or its frequency in the lot.

### A.2.2 Decision Variables

- $x_{bt} \in \{0,1\} \quad \forall b \in B,\ t \in \{1,\dots,m\}$: Binary variable representing the number of bales of class $b$ assigned to mixing table $t$.
- $y_t^i \in \{0,1,\dots,\alpha\} \quad \forall i \in T,\ t \in \{1,\dots,m\}$: integer variable representing the number of bales of cotton type $i$ assigned to mixing table $t$.
- $z_t^c \in \{0,1,\dots,\alpha\} \quad \forall c \in C,\ t \in \{1,\dots,m\}$: integer variable representing the number of bales of color grade $c$ assigned to mixing table $t$.
- $w_t \in \mathbb{R}^+ \quad \forall t \in \{1,\dots,m\}$: continuous variable representing the average Mic value of the bales allocated to mixing table $t$.

### A.2.3 Constraints

- Bale availability constraints:

$$\sum_{t=1}^{m} x_{bt} \le b_f \qquad \forall b \in B \tag{1}$$

- Exactly $\alpha$ bales per mixing table:

$$\sum_{b \in B} x_{bt} = \alpha \qquad \forall t \in \{1,\dots,m\} \tag{2}$$

- Linking type counting variables:

$$y_t^i = \sum_{\substack{b \in B \\ b_t = i}} x_{bt} \qquad \forall i \in T,\ \forall t \in \{1,\dots,m\} \tag{3}$$

- The maximum difference in the number of bales for each cotton type is at most 1:

$$y_{t_1}^i - y_{t_2}^i \le 1 \quad \forall i \in T,\ \forall t_1, t_2 \in \{1,\dots,m\},\ t_1 \ne t_2 \tag{4}$$

- Linking color grade counting variables:

$$z_t^c = \sum_{\substack{b \in B \\ b_c = c}} x_{bt} \qquad \forall c \in C,\ \forall t \in \{1,\dots,m\} \tag{5}$$

- The maximum difference in the number of bales for each cotton color grade is at most 1:

$$z_{t_1}^c - z_{t_2}^c \le 1 \quad \forall c \in C,\ \forall t_1, t_2 \in \{1,\dots,m\},\ t_1 \ne t_2 \tag{6}$$

- Average Mic per mixing table:

$$w_t = \frac{1}{\alpha} \sum_{b \in B} x_{bt}\, b_m \qquad \forall t \in \{1,\dots,m\} \tag{7}$$

- Breaking symmetric constraints:

$$w_t \ge w_{t+1} \quad t \in \{1,\dots,m-1\} \tag{8}$$

### A.2.4 Objective

Formally, the objective is: min $(w_1 - w_m)$.

## A.3 Pseudocodes of Constraint-Based Local Search for the BAPD Problem

The CBLS approach for the BAPD problem consists of three key elements, which were introduced in Section 4. The CBLS approach has the input described in Subsection A.2.1. Pseudocodes for the elements are presented in the following three subsections.

### A.3.1 Pseudocode of Random Initial Solution Generation

The pseudocode for generating a random initial solution is presented in Algorithm 1.

**Algorithm 1** CBLS: Random Initial Solution Generation.

$assignment[t][s] \leftarrow 0$ for every mixing table $t (t \in T)$ and slot $s (s \in \{1, \ldots, \alpha\})$;<br>
$slotCount[t] \leftarrow 0$ for every mixing table $t (t \in T)$;<br>
$typeCount[t][type] \leftarrow 0$ for every mixing table $t (t \in T)$ and type $type$;<br>
$colorCount[t][c] \leftarrow 0$ for every mixing table $t (t \in T)$ and color grade $c$;<br>
$maxTypeCount[type] \leftarrow \lceil \frac{\sum_{b \in B, b_t = tp} b_f}{m} \rceil$ (maximum number of bales with type $type$ in a mixing table);<br>
$maxColorCount[color] \leftarrow \lceil \frac{\sum_{b \in B, b_c = color} b_f}{m} \rceil$ (maximum number of bales with color grade $color$ in a mixing table) ;<br>
$T \leftarrow$ Shuffle $T$; $C \leftarrow$ Shuffle $C$;<br>
**for** $type \in T$ **do**<br>
    **for** $color \in C$ **do**<br>
        $numBale \leftarrow \sum_{b \in B, b_t = type, b_c = color} b_f$;<br>
        **while** $numBale > 0$ **do**<br>
            $t \leftarrow$ The mixing table satisfies (1) $typeCount[t][type] < maxTypeCount[type]$; (2) $colorCount[t][color] < maxColorCount[color]$; (3) $slotCount[t] < \alpha$; and (3) , among such mixing tables, prioritizes minimizing first $typeCount[t][type]$, second $colorCount[t][color]$, and last $slotCount[t]$;<br>
            $assignment[t][slotCount[t]] \leftarrow$ A bale with color grade $color$ and type $type$;<br>
            $slotCount[t] \leftarrow slotCount[t] + 1$;<br>
            $numBale \leftarrow numBale - 1$;<br>
            **if** $slotCount[t] = \alpha \quad \forall t \in \{1, \ldots, m\}$ **then**<br>
                **return** $assigment$<br>
            **end**<br>
        **end**<br>
    **end**<br>
**end**

### A.3.2 Pseudocode of Stage 1 – Finding Feasible Solution

The pseudocode for finding a feasible solution is presented in Algorithm 2.

**Algorithm 2** CBLS – Stage 1: Finding a feasible solution.

```
S ← random initial solution;
bestSol ← S;
Initialize tabu list, tbl, stable ← 0;
while getViolation(S) > 0 do
    S1 ← The neighbor of S in the N1(S) with lowest violation level by performing a substitution
      in mixing table t1 that is currently not in Tabu list;
    S2 ← The neighbor of S in the N2(S) with lowest violation level by performing a swap in two
      mixing tables t2 and t3 that are currently not in Tabu list;
    if getViolation(S1) ≤ getViolation(S2) then
        S ← S1;
        Add t1 to Tabu list;
    else
        S ← S2;
        Add t2, t3 to Tabu list;
    end
    if S is better than bestSol then
        bestSol ← S;
        stable ← 0;
    end
    else
        stable ← stable + 1;
    end
    // Adaptive Tabu tenure and restart mechanism (simplified)
    if no improvement was made in this iteration then
        Update tbl (increase if plateau, decrease if progress);
        if stable ≥ stableLimit then
            Restore best known solution or perform strong perturbation;
            stable ← 0;
        end
        if it mod restartFreq = 0 then
            Restart from a new random (or perturbed) feasible solution;
            Clear Tabu list;
            Reset tbl to middle value;
        end
    end
    it ← it + 1;
end
return S
```

### A.3.3 Pseudocode of Stage 2 – Improving Obtained Feasible Solution

The pseudocode for improving a feasible solution is presented in Algorithm 3.

**Algorithm 3** CBLS – Stage 2: Improving the objective of feasible solutions.

```
it ← 0;
bestSol ← S;
Initialize Tabu list, tbl, stable ← 0;
while it < maxIt do
    t+ ← table with highest average Micronaire value (MIC^max);
    t− ← table with lowest average Micronaire value (MIC^min);
    Find best allowed substitution in t+ decreasing max–min MIC gap most (Δ1);
    Find best allowed substitution in t− decreasing max–min MIC gap most (Δ2);
    Find best allowed swap between t+ and t− decreasing gap most (Δ3);
    if Δ1 ≥ Δ2 and Δ1 ≥ Δ3 and Δ1 > 0 then
        Perform substitution in t+;
        Add t+ to Tabu list;
    else
        if Δ2 ≥ Δ3 and Δ2 > 0 then
            Perform substitution in t−;
            Add t− to Tabu list;
        else
            if Δ3 > 0 then
                Perform swap;
                Add both t+ and t− to Tabu list;
            end
        end
    end
    if S is better than bestSol then
        bestSol ← S;
        stable ← 0;
    end
    else
        stable ← stable + 1;
    end
    // Adaptive Tabu tenure and restart mechanism (simplified)
    if no improvement was made in this iteration then
        Update tbl (increase if plateau, decrease if progress);
        if stable ≥ stableLimit then
            Restore best known solution or perform strong perturbation;
            stable ← 0;
        end
        if it mod restartFreq = 0 then
            Restart from a new random (or perturbed) feasible solution;
            Clear Tabu list;
            Reset tbl to middle value;
        end
    end
    it ← it + 1;
end
return bestSol
```