Proposed Solutions to the Tripper Car Positioning Problem

Felipe Novaes Caldas[1] and Alexandre Xavier Martins[2]
[1]*Gerência de Automação, Vale, São Gonçalo do Rio Abaixo, MG, Brazil*
[2]*Instituto de Ciências Exatas e Aplicadas, Universidade Federal de Ouro Preto, João Monlevade, MG, Brazil*

Keywords: Tripper Car, Combinatorial Optimization, Dynamic Programming.

Abstract: The trippers are equipments often found in mineral processing plants. Their role is to distribute ore coming from past stages of process in a silo with several hoppers. Positioning trippers is a scheduling problem defined by position determination of the equipment through the bins and along time. The system silo-tripper was modeled as a combinatorial linear optimization program aiming to get the optimal tripper positioning. Two paradigms were used to find out an exact solution: mixed integer linear programming and dynamic programming.

1 INTRODUCTION

Storage silos are structures designed to store bulk solid materials. Its function is to form an intermediary stock between two stages of the mineral processing cycle. Conveyor belts fulfill the role of bringing the ore from an earlier stage and storing it there. Afterwards, this material is removed by equipments called feeders, located right below each one of these chambers, proceeding with the mineral processing task (Wills and Napier-Munn, 2015). The number of feeders required in an intermediate storage silo design is proportional to the ore volume that will be handled. In this way, a silo must be subdivided into smaller chambers, according to the number of feeders required, providing an individualized area for the installation of each one of these equipments under the structure of the storage silo (Gupta and Yan, 2006). The vertical pillars that are observed along the lower structure of the building, as shown in Figure 1, indicate the approximated location of the subdivisions. The area defined by two pillars and filled by a concrete wall is the region where a chamber or bin of the silo is defined. In this example, there are sixteen subdivisions. In the right side of the image, there is a structure that projects over the front wall of the silo, it is called a conveyor belt. Its role is to carry ore coming from an earlier stage of the process to the storage silo.

A tripper is a car designed to distribute ore along the top opening of a storage silo (Wills and Napier-Munn, 2015). This equipment is constituted by a mobile metallic structure that physically supports a discharge point of the conveyor belt. The car is driven by steel wheels located under its structure. Metallic rails support and guide the tripper longitudinally along the silo, allowing the ore carried by the conveyor belt to be conveniently distributed between all subdivisions (Swinderman, 2014).

Figure 1: Example of storage silo from Brucutu mine – Vale.

An example of tripper can be seen in Figure 2. This equipment is located over the silo SL-133A-9101 at the processing plant from Brucutu mine. This site is an asset belonging to Vale mining company. The conveyor belt can be seen at the bottom left of the figure and runs all over the top of the storage silo. Its location is identified by the roller frame structures mounted along its side. The silo opening is shown below the conveyor belt and behind the guardrail, the ore is deposited in this region. The tripper is seen in the central area of the figure, over the conveyor belt. The discharge point extends from the top of the equipment and continues until reach the silo opening. One

Novaes Caldas, F. and Xavier Martins, A.
Proposed Solutions to the Tripper Car Positioning Problem.
DOI: 10.5220/0006806303440352
In *Proceedings of the 20th International Conference on Enterprise Information Systems (ICEIS 2018)*, pages 344-352
ISBN: 978-989-758-298-1

of the car rails is located to the right of the guardrail, it is supporting the tripper front right wheels.

Figure 2: Example of tripper car from Brucutu mine – Vale.

The flowchart shown in Figure 3 provides an illustrative example of a bulk mineral processing. The system's feed flow comes from a conveyor belt. The tripper has the role of distributing this material along the upper opening of the storage silo. The ore is withdraw by feeders and directed to sieves. In this stage, the material is split according to its granulometry, one part returning to the previous process as by-product 1 and another part proceeds further as by-product 2.

The tripper car movement is controlled by automation systems. Electric motors attached to the equipment wheels are driven by this systems (Boyer, 2010) in such way it is possible to position the equipment, defining which silo's bin will receive the material loaded by the conveyor belt. Three actions are possible through the automation operation interface: move to the right, move to the left, and halt. The ore flow isn't interrupted during the moving process.

Model-based predictive control (MPC) systems are a class of controllers that define their control action outputs based on future predictions of the process states. These predictions are performed by dynamic models that represent the behavior of the system over time, inside a finite time frame. As an optimizer, the MPC algorithm uses these informations to build an optimal control output aiming a desired behavior for the process. Only the present action of the entire output forecast window is sent to the plant at each iteration. In the next run cycle, the entire window is recalculated and the new control action is formed (García et al., 1989; Camacho and Bordons, 2007).

This first version of MPC was not able to handle discrete variables, restricting its application to continuous process. As an evolution of the initial proposal, the algorithm was expanded to consider discrete variables, logics and rules, in addition to the continuous variables that were already considered in the first version. This expanded form is known as hybrid MPC (Bemporad and Giorgetti, 2006; Borrelli et al., 2005). This controller is suitable to deal with the problem of tripper positioning since this process has both continuous (level, mass flow) and discrete (position) variables. A framework based on hybrid MPC was used by (Karelovic et al., 2015) to solve the tripper positioning problem. The criteria used by the authors to set up the controller objectives were bins level stabilization and minimization of the car movement.

The purpose of this work is to develop a solution to the tripper positioning problem based directly on mixed integer linear programming. In this case, the objective of the control system is to minimize the variations of bins levels, ensuring that these variables remain stable regarding the process constraints. Finally, a dynamic programming algorithm is proposed aiming the reduction of the time required to find the solution of the optimization problem.

2 DEVELOPMENT

The development of the mathematical models that describe the behavior of the silo-tripper system is the starting point for the study of the solutions to the problem of car positioning. The features of the optimization model, such as instance input data, decision variables, constraints and objective function, are described in detail. This formulation will be solved later by means of mixed integer linear programming and dynamic programming.

2.1 Mathematical Model

The Figure 4 presents the moving possibilities of a n-bins silo. At any position, the equipment can only move to one of the adjacent places or remain in the same position over which it is located. For example, if it's in p_2 position then the movement options will be $\{p_1, p_2, p_3\}$. Moving the car to a position not adjacent to the current one implies crossing all the intermediate positions between the points of departure and destination. Consequently, tripper directs the ore flow to the bin whose position is overlapped by current equipment displacement. The choice of feeding position is a task with temporal dependence since the movement speed of the equipment is finite.

There is a risk that the material may eventually lacks in one of the bins since tripper may only be in a particular place at any time. This may occur even in situations where the mass balance of the system is balanced, that is, the total mass flow fed is equal to the total withdrawn. To work around this problem, the tripper must be moved along the bins openings so that

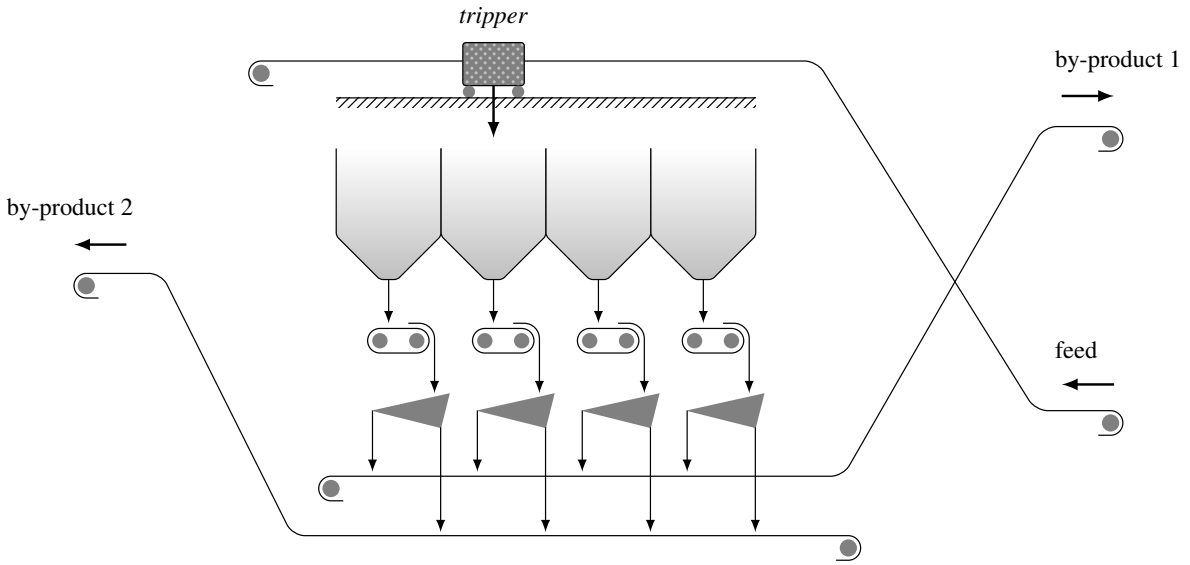

Figure 3: Diagram illustrating a granular ore sifting process. The tripper is represented by a dotted rectangle with two wheels indicated as small circles. A four-bin silo is shown under the tripper. Below each bin there is a belt feeder. The sieves are represented by gray triangles. Conveyor belts are signaled by the ore flow they carry: feed, by-product 1 and by-product 2.

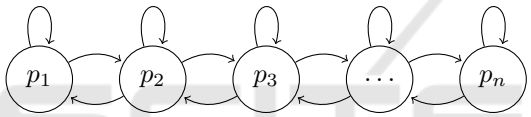

Figure 4: State machine representing the possibilities of tripper positioning.

the material addition by the conveyor belt does not allow any lack of ore. It is also needed to limit the bins levels according to the maximum capacity restrictions of the silo.

The time required for the tripper to travel through the region corresponding to the opening of a bin, moving uninterruptedly, is proportional to the length of the traveled region and inversely proportional to the moving speed of the car. In order to simplify the model, the time elapsed by tripper to cross a whole bin is considered constant in any situation in which the silo subdivisions have the same size and the tripper has constant speed. The forecast time window, over which the prediction will be performed, is divided into multiples of this minimum time. The size of this window defines the first dimension of the positioning problem model.

The second problem dimension is defined by the set P of feasible positions for the tripper, whose number of elements corresponds to the number of bins. The positioning scheduling is defined as the choice of a sequence of car movement actions along a finite window of future time. The purpose of these actions is to keep the variables controlled within the operational constraints established for the problem. The future time window is mapped to a finite set T, composed of the current state of the system added by $t-1$ future steps. These extra iterations are necessary to make the model able to predict in advance the outcome of the control actions that will be taken over time, verifying if the established constraints will be violated at some point.

The Figure 5 shows a graph of the model, developed to represent a tripper positioning problem. The initial positions, relative to the time t_1, are contained in the set $\{p_{1i} \in \mathbb{N} \mid 1 \leq i \leq n\}$, being n the number of bins. From any initial position, the subsequent positions are chosen at each iteration, regarding the movement constraints over which the possible positions are limited the current position and adjacent ones. Finally, the possible positions in the final step t_t are contained in the set $\{p_{ti} \in \mathbb{N} \mid 1 \leq i \leq n\}$.

Therefore, the parameters defining the size of an instance are:

- Number of bins: n;
- Prediction window size: t.

2.1.1 Instance Parameters

The instance parameters are presented in the Table 1. In this work, the input rates q and output set Q are defined as constants throughout all steps. However, this parameters may be easily transformed to have multiple different values through time.

The instance size is defined by the sets T periods

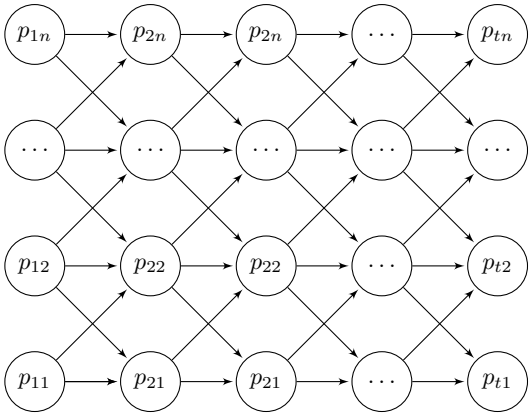

Figure 5: Graph representing the positioning possibilities of a tripper throughout iterations.

of time and P bins. The initial level of each one of the bins is defined by the vector L_1. The levels values are constrained between $l_{\min}$ and $l_{\max}$. The starting position of the tripper car is defined by p. The silo emptying rate is defined as K. This parameter defines the amount of ore that each one of the bins is capable of storing according to the dynamic equation of the process.

2.1.2 Decision Variables

The decision variables are presented in the Table 2. The most important of them is the matrix **I** that represents the tripper positions through time. It is a set of binary values that defines the car positioning over n possible positions along t steps (Wolsey, 1998).

The states of the P bins levels along the T iterations are defined by **L**. The matrices **A** and **B** represent slack variables for bins levels. The Z set represents the auxiliary variables for the *MinMax* cost function.

Table 1: Instance parameters.

Parameter	Description
T	Periods set
P	Bins set
n	Number of bins
t	Number of periods
L_1	Inicial level
$l_{\max}$	Maximum level
$l_{\min}$	Minimum level
q	Input mass flow
Q	Output mass flow
K	Bins level coefficient
p	Initial position

Table 2: Decision variables.

Variable	Description
I	Tripper positions
A	Low level slack
B	High level slack
L	Levels
Z	*MinMax* auxiliary

2.1.3 Constraints

The dynamic equation that defines the level of a bin i along time is shown by Equation 1 (Pal et al., 2014):

$$L_i = K_i \int (Q_t - Q_i)dt \tag{1}$$

Being L_i the level of some bin, K_i the bin volume coefficient and Q_t the mass flow of the tripper and Q_i of an individual feeder.

The Equation 13 defines the mass accumulation at each period of time based on Equation 1, adding the slack variable components. At each step, the bin located under the current tripper position is increased by the mass flow input q. The initial level is defined by Equation 8. This constraint represents the available ore mass in the bin at first period of time.

If the Equation 1 is not balanced, for example, the input mass rate is different from the mass output rate, the bin may empty or fill up completely occasionally in any of the solution steps. Two slack variables sets **A** and **B** have been added to ensure that the problem becomes feasible with any instance parameters. The first handle bin level below $l_{\min}$, as defined in Equations 11 and 12, and the second handle bin level above $l_{\max}$, as shown by Equations 9 and 10.

The tripper starting position is defined by parameter p in Equation 7. The Equation 3 determines that the car may be only on a single bin at each iteration. The moving rules along the periods of time are defined by the Equation 4. It defines that the car position in a given iteration is only feasible if the tripper is kept at the same place or moves to a immediately adjacent position. The Equations 5 and 6 determine the moving rules if the equipment is in the first or last position over the silo.

$$Z_j \leq \mathbf{L}_{i,j},\ \forall\ j \in T,\ i \in P \tag{2}$$

$$\sum_{i \in P} \mathbf{I}_{ij} = 1,\ \forall\ j \in T \tag{3}$$

$$\mathbf{I}_{i,j} \leq \mathbf{I}_{i-1,j-1} + \mathbf{I}_{i,i-1} + \mathbf{I}_{i+1,j-1},\quad \forall\ i \in [2, n-1],\ j \in [2,t] \tag{4}$$

$$\mathbf{I}_{1,j} \leq \mathbf{I}_{1,j-1} + \mathbf{I}_{2,j-1},\ \forall\ j \in [2,t] \tag{5}$$

$$\mathbf{I}_{n,j} \leq \mathbf{I}_{n,j-1} + \mathbf{I}_{n-1,j-1},\ \forall\ j \in [2,t] \tag{6}$$

$$\mathbf{I}_{p,1} = p \tag{7}$$

$$\mathbf{L}_{i,1} = \mathbf{L}_1,\ \forall\ i \in T \tag{8}$$

$$\mathbf{A}_{i,j} = \Delta\mathbf{A}_{i,j} + \mathbf{A}_{i-1,j},\ \forall\ i \in P, j \in T \tag{9}$$

$$\Delta\mathbf{A}_{i,1} = 0, \forall\ i \in V \tag{10}$$

$$\mathbf{B}_{i,j} = \Delta\mathbf{B}_{i,j} + \mathbf{B}_{i-1,j},\ \forall\ i \in P, j \in T \tag{11}$$

$$\Delta\mathbf{B}_{i,1} = 0, \forall\ i \in V \tag{12}$$

$$\mathbf{L}_{i,j+1} = \mathbf{L}_{i,j} + K_i(q \cdot \mathbf{I}_{i,j} - Q_i) + \Delta\mathbf{A}_{i,j+1} - \Delta\mathbf{B}_{i,j+1},\ \forall\ i \in P, j \in [1, t-1] \tag{13}$$

$$\mathbf{L}_{i,j} \leq l_{\max},\ \forall\ i \in P, j \in T \tag{14}$$

$$\mathbf{L}_{i,j} \geq l_{\min},\ \forall\ i \in P, j \in T \tag{15}$$

2.1.4 Objective Function

The objective function is based on the principle *Min-Max* (Rich and Knight, 1991). Its purpose is to maximize the lowest level value between all bins at each period, not allowing ore to lack in any bin at no time. The Z set, as shown by Equation 2, is defined as the lowest value of the bins levels reached on each step. The solution is found by maximizing the result of Z set summation. The cost function for the problem can be seen in Equation 16:

$$\text{Maximize} \sum_{j \in T} Z_j - \sum_{i \in P, j \in T} \mathbf{A}_{ij} - \sum_{i \in P, j \in T} \mathbf{B}_{ij} \tag{16}$$
$$s.t.\ Z_j \leq \mathbf{L}_{i,j},\ \forall\ j \in T,\ i \in P$$

2.2 Mixed Integer Linear Programming

The mixed integer linear programming is a mathematical method whose purpose is to find the best solution for a cost function within a set of constraints defined by linear relations. The variables handled by this formulation can be represented as either continuous or integer values(Papadimitriou and Steiglitz, 1998).

Two solvers are used to deal with the proposed model for the tripper positioning problem. The first one is the Glpk (GNU Linear Programming Kit). The second choice is IBM's Cplex Optimizer. The objective is to compare between the outcome of an open source solver relative to one proprietary system in task of solving the optimization problem.

2.3 Dynamic Programming

Dynamic programming is a paradigm of computer science characterized by find the solution of a complex problem by solving its smaller subproblems. Each subproblem is solved only once and the result is stored in a table to be query later when needed, rather than recursively computing it whenever necessary (Levitin, 2012; Cormen et al., 2001).

The subproblems of the positioning problem are defined as the solution of the this problem in each one of the intermediate steps. This procedure starts at the first iteration goes through incrementally until reach the last period. Two tables were created to store the partial result of the subproblems: the first handle cost values and the second handles positions history. These tables are identified as *Table* and *P* in the Algorithm 1. The variables referenced by this pseudocode are listed in Tables 1 and 2.

The algorithm first step is to update the bins levels L according to the dynamic model of the process (lines 2 and 3). In the next step, the value of the cost function c is updated with the lowest level value in the current iteration (line 4). The variable *Table* is updated if there is no cost value related to the current step recorded in *Table* or if the current cost is greater than or equal to the already stored value, then recursion is continued (line 6). Otherwise, if the current cost is less than the recorded cost, the function returns immediately. The best possible solution is stored in the global variable *Best* in the last step of the tripper positioning procedure (lines 12 and 14).

Algorithm 1: Dynamic Programming.

```
 1: function SEARCHMAXIMUM(c, L, Q, q, P, p, s)
 2:   L[p] ← L[p] + q
 3:   L ← L − Q
 4:   c ← c + min(L)
 5:   if Table[s] = ∅ or c ≥ Table[s] then
 6:     Table[s] ← c
 7:     P[s] ← p
 8:     if s ≠ t then
 9:       for i ← max(1, p − 1) to min(n, p + 1) do
10:         SearchMaximum(c, L, Q, q, i, P, s + 1)
11:       end for
12:     else
13:       if Table[t] < c then
14:         Best ← P
15:       end if
16:     end if
17:   end if
18: end function
```

The Figure 6 gives an example of the execution of the proposed algorithm for instance of three bin silo and four units time window. Each one of figures, ran-

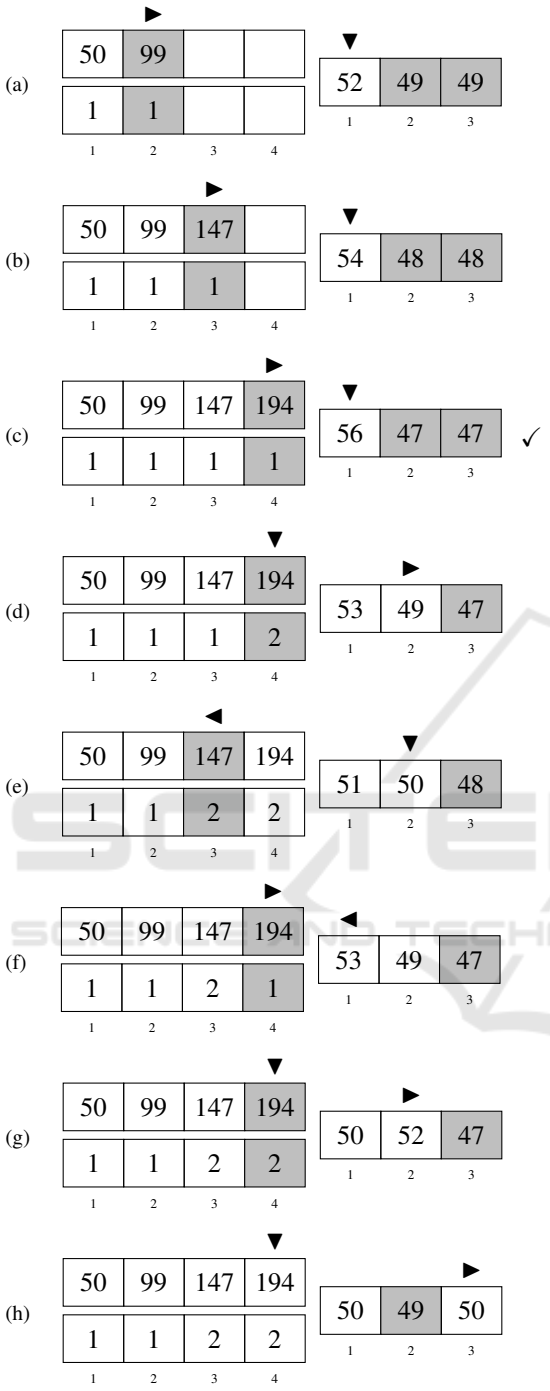

Figure 6: Example of eight iterations of dynamic programming algorithm for a silo with three bins.

ging from 5(a) to 5(h), represents one of the algorithm steps. The two tables on the left side of any subfigure represent *Table* and *P*, respectively. The arrow over them indicates the current period of time and the recursion direction. The left-sided arrow means that the algorithm execution is returning to a previous step, a right-arrow indicates that the execution is advancing to the next step and, finally, the down arrow shows that the position is at the same place as it were in last iteration. A gray filling color in a cell indicates an update in its value. The table on the right side of a subfigure represents the current bins levels. An upper arrow indicates the position and direction of the tripper movement. Minimum levels in each period are indicated by a gray color filling the respective cells.

The first step of execution is shown by Figure 5(a). The three bins are initialized with 50% level value, the tripper feed flow rate is set with 3 and the three feeders flow rate with 1. The first action of the car is to remain at the bin 1. The levels are decremented by one unit and the bin 1 level is added by 3 units. The bins with the lowest level value are 2 and 3, they are colored with gray. The current step cost is the initial cost 50 from the past step added by the minimum step level value 49, which is the lowest level value. The new partial cost for this step is 99. This value is stored in the table along with the position of the car.

This procedure is repeated until there are no new additions to the table or until the execution of the algorithm reaches the maximum limit of steps t. In this case, the current solution becomes the new best if its cost is higher than the cost of the previous best solution already found so far. A check mark label ✓ is added to a subfigure indicating that a new best solution was found, as shown in Figure 5(c).

3 RESULTS

Three tests are presented in this section. The first one is a comparison between the control proposal by level stabilization and minimization of the car movement, as indicated by (Karelovic et al., 2015), and the algorithm proposed in this work for the positioning problem. The second test is done over a silo with imbalanced flow rates, the purpose is to verify the proposed model response in this scenario. The last test compares the performance of the Cplex and Glpk resolvers with respect to the dynamic programming algorithm. The computational tests were performed over an Intel Core i5 4570 machine with 16GB of DDR3 memory.

The first test was performed over a three-bin silo instance. The initial levels values were defined as $L = \{90, 50, 10\}$, the feeder flow rates $Q = \{3, 1, 2\}$, the tripper feeding flow rate was $q = 6$ and the starting position, $p = 1$. The objective is to compare the two proposals of silo level control. The stabilization strategy was adjusted in way to keep bins levels within an acceptable range of $20\% \leq \mathbf{L} \leq 80\%$.

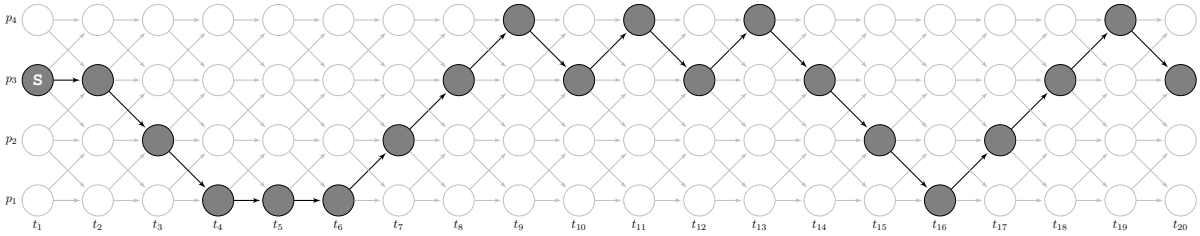

Figure 7: Positioning schedule for a tripper car. This instance is composed by a 4-bin silo with a time window of 20 periods. The white letter *S* indicates the starting position.

The Figure 8 shows the test results. The upper chart shows the behavior of the bins levels values over time in response to the actions taken by tripper to maximize the lowest bin level in each iteration. In this case, after approximately 20 time periods, the levels stabilize within a value range ranging from 40% to 60%. The test of movement minimization strategy, as seen in the lower chart of Figure 8, could kept the bins levels within the requested value range of 20% and 80%. In this case, the number of car movement were much smaller than in the first strategy, as indicated by the small amount of reversions in the rate of variation of the levels values.

The test was done considering an instance with unbalanced loads. The system was set up as a 4-bins silo, where feed output flow rates were $\{Q_i = 4, \forall\, i \in P\}$ and input flow rate $q = 16$, the initial levels were defined as $L = \{50, 60, 30, 60\}$. In this case, there is a real possibility of material lacking in some of the bins if the tripper positioning scheduling is done incorrectly. Figure 7 shows the resulting positioning for this instance. The white letter *S* indicates the start position $p = 3$. The car is positioned through the 20 periods in all possible positions in order to maximize the minimum levels at each period over time.

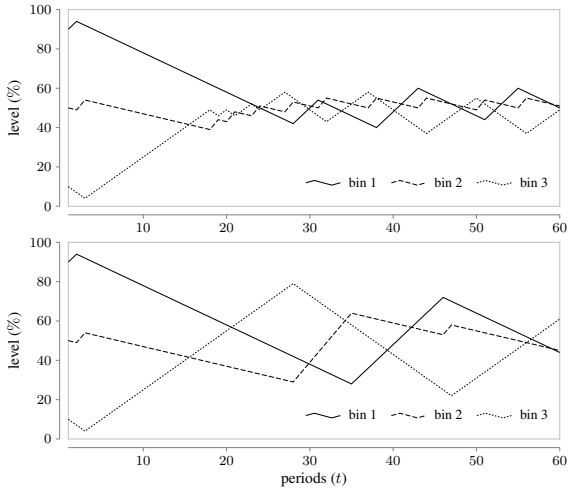

Figure 8: Comparative tests between the strategies of movement minimization (top chart) and maximization of the lowest level at each iteration (bottom chart).

The imbalance between input and output flow rates implies in decline of the bins average levels through time as shown by Figure 9. The initial and final average levels were 50 and 31 confirming the expected behavior. However, the initial standard deviation of bins levels was 14.1 against the final value of 6.6, that is, not only the minimum levels were kept at higher values but their variabilities was also decreased.

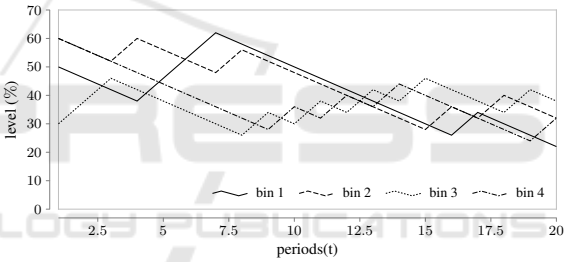

Figure 9: Results of a test with a 4-bin silo instance in a 20 periods time window.

The purpose of the third set of tests is to compare the performance of the solvers and the dynamic programming algorithm. The tests were done in a silo instance with $n \in \{3, 5, 9, 16\}$ bins and with a forecast window of t time periods. These parameters were varied exhaustively within a time range of 1000 seconds, to verify the response time of the solvers. The parameters of flow rates chosen for the feeders were $\{Q_i = 1, \forall\, i \in P\}$, the feed flow rate of the tripper was $q = n$ and the starting position was $p = 1$. It is expected that the time elapsed to solve an instance is proportional to the number of bins n. This variable represents one of the problem sizes, along to the time periods t, and its increase is directly related to an increase of the problem complexity.

The Figure 10 displays the tests results. The dynamic programming algorithm was asymptotically faster in tests with the number of bins $n \in \{3, 5, 9\}$. The sharp discrepancy between the result of DP regarding to the results of solvers in the tests with $n \in \{5, 9\}$

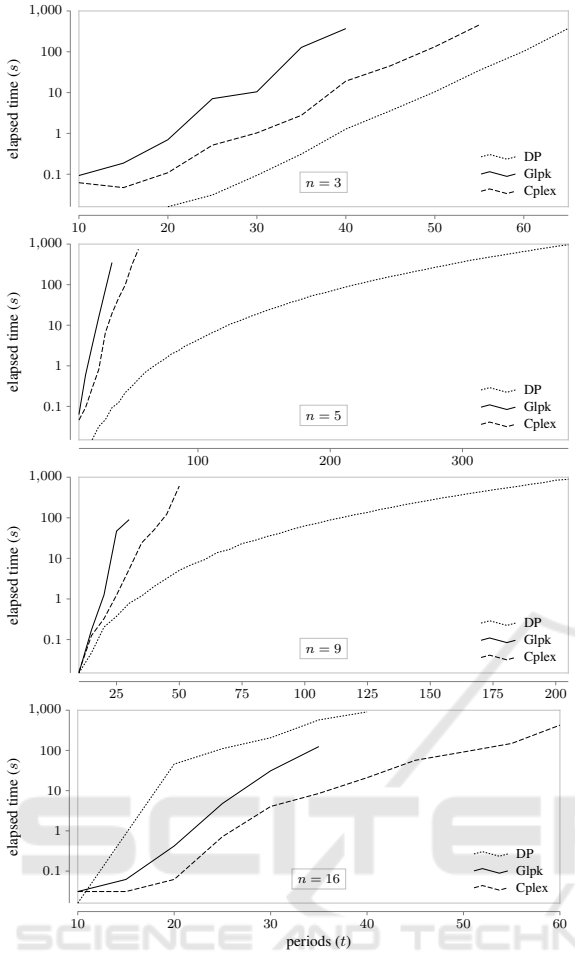

Figure 10: Tests results of Cplex and Glpk solvers and dynamic programming algorithm.

suggests that in these cases the dynamic programming has order of complexity lower than the algorithms of mixed integer linear programming. With $n = 16$ the two solvers results were superior to the proposed algorithm. At $n = 3$, although DP performance was faster than the other methods, it was asymptotically similar. One theory to explain this behavior is that the algorithm could not take advantage of higher level of symmetry in a $n = 3$ instance. Cplex was faster than Glpk in all instances tested. One point in favor of the dynamic programming algorithm is the fact that Cplex is able to handle multiple threads (were four in this scenario) boosting performance of solving process. While current implementation of the proposed solution presented in this work can just handle one thread at time.

4 CONCLUSION

This work proposes a new approach to solve the tripper positioning problem. The objective was to model the silo-tripper system as a combinatorial optimization problem and solve it by means of exact methods directly, without the use of hybrid MPC frameworks. The proposal was to maximize the lowest bin level of the silo in each period, in order to achieve maximum stability in the tripper operation. This strategy was compared with another option indicated in the literature by (Karelovic et al., 2015). In this case, the author propose to minimize the amount of car movement and the stabilization of bins levels values. Algorithms of mixed integer linear programming and dynamic programming were used to handle the problem.

The algorithm proposed by this work maximized the stabilization of bins levels values, taking advantage of the freedom of car movement to minimize level variations more intensely than the stabilization algorithm. In addition, the performance of the dynamic programming was superior to the the mixed integer linear programming solvers in the tests in the instances set featuring bins number lower than $n = 16$, and, in some cases, the dynamic programming shown order of complexity suggestively inferior to the other algorithms within the time window evaluated. However, there was an inconsistence in the 3-bin instance DP results that shown the same asymptotic behavior as the solvers. This issue is beyond the scope of this study and will de addressed in a future work.

The next steps of this research will focus on finding alternative methods to solve the tripper positioning problem. The goal will be to find metaheuristics capable of finding good or even optimal solutions, but with search time significantly lower than the exact methods.

REFERENCES

Bemporad, A. and Giorgetti, N. (2006). Logic-based methods for optimal control of hybrid systems. *IEEE Transactions on Automatic Control*, 41(6):963–976.

Borrelli, F., Baotic, M., Bemporad, A., and Morari, M. (2005). Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. *Automatica*, 41(10).

Boyer, S. A. (2010). *SCADA: Supervisory Control and Data Acquisition*. International Society of Automation, 4 edition.

Camacho, E. F. and Bordons, C. (2007). *Model Predictive Control*. Advanced Textbooks in Control and Signal Processing. Springer-Verlag London, 2 edition.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms*. McGraw-Hill, 2 edition.

García, C. E., Prett, D. M., and Morari, M. (1989). Model predictive control: theory and practice – a survey. *Automatica*, 25(3):335–348.

Gupta, A. and Yan, D. (2006). *Mineral Processing Design and Operation*. Elsevier Science, 1 edition.

Karelovic, P., Putz, E., and Cipriano, A. (2015). A framework for hybrid model predictive control in mineral processing. *Control Engineering Practice*, 40:1–12.

Levitin, A. (2012). *Introduction to the design & analysis of algorithms*. Pearson, 3 edition.

Pal, B., Sana, S. S., and Chaudhuri, K. (2014). A multi-echelon production-inventory system with supply disruption. *Journal of Manufacturing Systems*, 33:262–276.

Papadimitriou, C. H. and Steiglitz, K. (1998). *Combinatorial Optimization: Algorithms and Complexity*. Dover, 1 edition.

Rich, E. and Knight, K. (1991). *Artificial Intelligence*. McGraw-Hill Inc., 1 edition.

Swinderman, R. T. (2014). *Belt Conveyors for Bulk Materials, w/Metric Conversion*. Conveyor Equipement Manufactures Association, 7 edition.

Wills, B. A. and Napier-Munn, T. (2015). *An Introduction to the Practical Aspects of Ore Treatment and Mineral Recovery*. Butterworth-Heinemann, 8 edition.

Wolsey, L. A. (1998). *Integer Programming*. Wiley-Interscience.