Asynchronous Team for Flow Shop Scheduling Problem: A Case Study

Luis Enrique Tarazona Torres
Department of Industrial Engineering
Universidad de los Andes
Bogota, Colombia
le.tarazona@uniandes.edu.co

Ciro Alberto Amaya Guio
Department of Industrial Engineering
Universidad de los Andes
ca.amaya@uniandes.edu.co

David Álvarez-Martínez
Department of Industrial Engineering
Universidad de los Andes
d.alvarezm@uniandes.edu.co

Abstract

Mining companies that carry out material movement activities for the heap leaching process maintain a constant sequence of operations. Properly scheduling these operations is crucial to reduce operating costs. The scheduling of mining activities can be modeled as a flow shop problem. The efficient scheduling of mine operations must be continuously monitored to ensure positive impacts. Implementing technologies such as Digital Twin for process planning and control in industries such as mining is now expected. This paper presents a methodology based on an asynchronous team of metaheuristics to solve the flow shop problem resulting from a case study of a Chilean mining company. Tabu search algorithms were used as team members. An asynchronous team was selected because it maximizes the use of computational resources, thus providing the Digital Twin with a sophisticated optimization algorithm for operation scheduling. The performance of the asynchronous team was validated by benchmarking against the best works published in the literature. A set of classical instances was used due to the company's limitations in using their information. As future work, it is expected to contrast the improvements in planning and control of mining activities with the asynchronous team embedded in the Digital Twin.

Keywords
Asynchronous team, Digital Twin, Flow shop problem, Mining activities, Scheduling

1. Introduction
Within the mining industry, leach material movement activities maintain a constant sequence of operations; the essential factors in scheduling these operations are operating time, resource availability and utilization, and production costs. Optimal scheduling of operations (production scheduling) can help ensure that this process can respond quickly to changes in work demand and meet assigned targets (Peña et al. 2022). The scheduling of material movement and site preparation activities in the heap leaching process is crucial to ensure the efficiency and profitability of the production process. Several factors must be considered to carry out this task, such as the availability of machinery, the type of ore to be processed, and weather conditions (Checya 2015). Within mines, it is essential to have rigorous planning to ensure that resources are available at the right time and in the necessary quantity (Barrios 2020).

Efficient scheduling must be accompanied by prior and continuous control of operations; the use of technologies for control and monitoring in the mining sector has increased, given the significant savings obtained and their ability to mitigate major risks. One technology that has demonstrated this is the Digital Twin. Specifically, it has been applied to material loading activities in mining processes (Tliba et al. 2022). The Digital Twin is a technology that allows the creation of a virtual model of a process or system in real-time, letting monitoring and simulation of future events (Eunike et al. 2022). This technology simulates different scenarios and evaluates their impact on the process, allowing informed decisions to be made. For example, simulations can be performed to evaluate the impact of weather conditions, machinery availability, and other factors that may affect the process (Hazrathosseini and Afrapoli 2023).

Production scheduling is a critical topic in operations management and refers to planning the resources required to produce goods or services. The flow shop problem is a classical production scheduling problem that has been the subject of intense research in the literature. The flow shop problem is one of the most studied problems in this area and is characterized by having several jobs to be processed in a given sequence by a series of machines. The main objective is commonly to minimize the total processing time of all the jobs (*Cmax*), although there may be other objectives to be considered. This problem is called the permutation flow shop problem (PFSP); and is named F/prmu/Cmax (Pinedo 2016). Due to its computational complexity, the flow shop problem is a problem considered NP-hard in the scheduling field. The movement of material for the heap leaching process can be considered an instance of the flow shop problem since the process is sequential, performing activities for material feeding and land preparation for heap assembly. The objective is to optimize available resources, such as processing time and the number of machines used, to minimize costs and improve production efficiency.

The flow shop problem has been solved using exact and approximate optimization techniques. Exact techniques guarantee optimal solutions, but their computational complexity increases rapidly with the size of the problem (Pinedo 2016). On the other hand, heuristic techniques seek feasible solutions of good quality in a reasonable time. Specifically, metaheuristics have been widely used due to their excellent results for large problem instances (Talbi 2009). Asynchronous metaheuristic teams allow maximum use of computational architectures with multiple processors and large memory (RAM) and achieve excellent results by combining the qualities of different metaheuristics.

In this work, we present an asynchronous team of metaheuristics (ATTS) to solve the flow shop problem of a Chilean mining company. The metaheuristic is embedded in a Digital Twin model, thus taking advantage of the available computational architecture. The efficiency of the proposed asynchronous team is tested using a set of instances presented in the literature due to research publication conflicts given the confidentiality clauses imposed by the Chilean company. 90 instances proposed by Taillard (1990) were used, which have up to 100 jobs in 20 machines, thus testing the performance of the proposed asynchronous team in problem sizes ten times larger than that of the mining company.

The results obtained by the asynchronous team are acceptable in terms of quality and computation time, and computational resources are used to the maximum, providing the Digital Twin with a sophisticated optimization algorithm for scheduling operations, which in the future will be reflected in the minimization of production costs for the company.

2. Problem Description and Literature Review

The case study evaluates a heap-leaching process for recovering copper from riprap dumps. The heap-leaching process is sequential; it starts its operations with the assembly of the mixed heaps; for this, the green riprap (M1) and red riprap (M2) must be moved to the mixed heap assembly areas (M3), then the mixed riprap is transported to a dump (M4), before entering the dump the material goes through a sieve to remove large particles and other residues (M5). Conveyor belts transport the material to an agglomeration zone (M6), where the size of the material is homogenized. The heap assembly areas are conditioned to receive the material (M7). Belts transport this agglomerated material to the leaching heap areas to build the heaps (M8). The heaped material is irrigated with sulfuric acid (M9). The acid captures the copper and is drained from the lower part of the heap and transported to the copper recovery plant. The leached riprap is removed and transported to the mine dumps (M10). The process has a total of ten operations, each one considered as a machine (M).

On the other hand, currently at the mine, there are eight green riprap zones, two red riprap zones, four mixing pile zones, the unloading box, the agglomeration zone, land preparation, leaching pile construction, leached riprap removal and the copper recovery zone. Since the work cannot be carried out in parallel due to the availability of machinery, each job starts operations in one of the green riprap zones, thus generating different processing times in the different machines. Figure 1 shows the location of most of the machines and the green riprap zones.

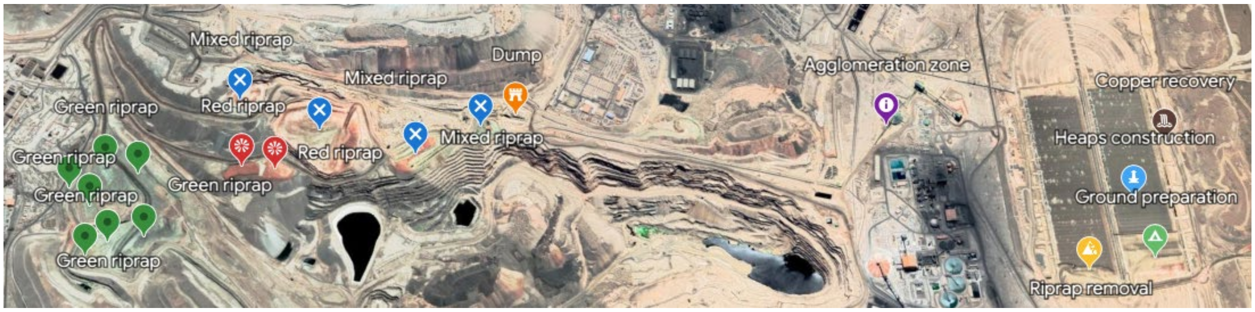

Figure 1. Distribution of machines and work in the mine.

In this paper, we reviewed (meta)heuristic algorithms proposed for the flow shop problem seeking to minimize the makespan (*Cmax*), especially the parallel processing implementations. Early works of Johnson (1954), Campbell et al. (1970), and Dannenbring (1977) proposed constructive heuristics for exceptional cases of the flow shop problem. Nawaz et al. (1983) presented a constructive heuristic method based on a rule of selecting jobs that gives a bigger priority (program first) to the jobs with a higher total process time. Suliman (2000) presented a local search heuristic using a transition mechanism based on exchanging a pair of jobs following a rule of directionality.

Among the metaheuristics, we highlight the tabu search algorithm proposed by Ponnambalam et al. (2000) and Taillard (1994), the hybrid algorithm between the genetic algorithm and the variable neighborhood search by Pandolfi et al. (2009), the variable neighbor search algorithm proposed by Zobolas et al. (2009), the Particle Swarm Optimization algorithm of Marinakis & Marinaki (2013), recently the hybrid evolutionary algorithm proposed by Khurshid et al. (2021). Finally, the iterative algorithm presented by Ruiz & Stutzle (2009) that uses a simulated annealing scheme to accept worse-quality solutions is highlighted.

On the other hand, reviewing the Digital Twin technology together with the flow shop problem and its application in mining, we find works such as Tliba et al. (2022), where to optimize, they developed a mixed integer linear programming (MILP) model and simulated a 3D shop floor model including stochastic aspects and the additional constraints that are difficult to model in MILP; the digital twin they proposed can reschedule production activities based on internal and external events. Eunike et al., (2022) developed an architecture to build a decentralized scheduling system resilient to disruptions on the shop floor; in this case, they developed a combined local scheduling and contract network protocol strategy to improve production scheduling. Hazrathosseini & Afrapoli (2023) conducted a study that showed a more comprehensive overview of simulation applications in open pit mining and proposed an exemplary six-layer digital twin architecture for this type of mine. Peña et al. (2022) developed a digital twin simulation based on discrete event simulation (DES) and a learning model to incorporate geological variation for its uncertainty.

3. Methods

The Flow-shop problem can be mathematically formulated as a minimization problem as in Equations 1-6 (Pinedo 2016).

Parameters, indexes and sets:
n: Number of jobs
m: Number of machines
j: Index of jobs, $j = 1 \dots n$
i: Index of machines, $i = 1 \dots m$
J: Set of jobs
M: Set of machines
p_{ji}: Processing time of job j in machine i

s_{ji}: Start time of job j in machine i

$$M_{ji} = \begin{cases} 1 \text{ if job } j \text{ is processing in machine } i \\ 0 \text{ otherwise} \end{cases}$$

Variables

$$x_{jk} = \begin{cases} 1 \text{ if job } j \text{ is the position } k \text{ in the sequence} \\ 0 \text{ otherwise} \end{cases}$$

Minimize:

$$Z = Cmax \tag{1}$$

Subject to:

$$\sum_{j=1}^{n} x_{jk} = 1 \quad for\ k\ \in J \tag{2}$$

$$\sum_{k=1}^{n} x_{jk} = 1 \quad for\ j\ \in J \tag{3}$$

$$\sum_{k=1}^{n} x_{jk+1}(p_{ji} + s_{ji}) \times M_{ji} - \sum_{k=1}^{n} x_{jk}(p_{ij+1} + s_{ji+1}) \times M_{ji+1} \tag{4}$$

$$\sum_{k=1}^{n}\sum_{i=1}^{m} x_{jk}(p_{ji} + s_{ji}) \times M_{ji} \leq C_{max} \tag{5}$$

$$x_{jk}\ \in \{0,1\}\ j,k\ \in J \tag{6}$$

Equation 1 shows the objective function, which is to minimize the total job processing time (Cmax). Equation 2 ensures that each position has only one associated job, and Equation 3 indicates that each job is in one position. Equation 4 defines the difference between the time when the job at position *k+1* starts at machine *i+1* and the time when job *k* ends at a machine *i*. Equation 5 captures the total completion time Cmax. Equation 6 sets the nature of the variables.

Due to the high computational cost of the solution of the mathematical model, an approximate approach was chosen in this work. Specifically, it is intended to take advantage of the computational architecture that holds the Digital Twin. Therefore, an algorithm that uses parallel processing has been selected, especially following an asynchronous processing philosophy; in addition to this, inspired by Ruiz & Stutzle (2009), a basic metaheuristic algorithm is presented that through small perturbations and a random exploration of the neighborhood manages to achieve good quality solutions in reasonable computation times. To the best of our knowledge, in the literature, there is no asynchronous metaheuristic team for the flow shop problem (Crainic & Toulouse, 2010); this motivated us to present an asynchronous team of tabu search algorithms.

Asynchronous team (AT) architecture offers three key advantages: modularity, distribution, and robustness. In an AT, the members are autonomous, and there is no dependency between them for execution or communication. Therefore, in the software development process, members can be built independently, allowing the implementation of the solution to be modularized. The asynchronous nature of ATs makes them suitable for parallel processing. System reliability is critical mainly when the methodology is used in production environments (as in Digital Twin). ATs are robust because the failure of one member does not bring down the entire system. If, under circumstances, a member produces an invalid solution or fails while generating a solution; this will not affect the operation of the other members.

In asynchronous teams, the abilities to solve a problem (optimization algorithms) are packaged as members. Members can, and usually should, be diverse; some may iterate faster while others may be slow; some may make minor improvements while others may make radical changes. The members in ATs are identical in two aspects, they are all completely autonomous (they work without supervision), and they all have the same work cycle (in each iteration, each member performs the same three steps: perturb the best solution found; try to improve it and finally, update the best solution if applicable).

Algorithm 1. Asynchronous team of metaheuristics for the flow shop problem (AT_{TS})

Input: *num_jobs:* Number of jobs to be perform*, num_machines:* Number of machines in the sequence, *processing_times;*
Parameters: MAX_GLOBAL: total number of global iterations, MAX_TEAMS: total number of asynchronous metaheuristic algorithms, MAX_LOCAL: total number of local iterations, MAX_NEIGBORS: total number of neighbors randomly explored in the neighborhood;
Output: *Incumbent:* best scheduling found during the optimization algorithm

```
Initial_Solution ← NEH_ConstructiveAlgorithm(num_jobs, num_machines, processing_times)
Shared_Best_Solution ← Initial_Solution
Incumbent ← Initial_Solution
For i ← 1 to MAX_GLOBAL
    For j ← 1 to MAX_TEAMS
      CreateAsynchronousParallelProcess(Shared_Best_Solution)
      Local_Best_Solution ← Tabu_Search(Shared_Best_Solution; MAX_LOCAL, MAX_NEIGBORS)
      If Makespan(Local_Best_Solution) < Makespan(Shared_Best_Solution) Then
           Shared_Best_Solution ← Local_Best_Solution
    If Makespan(Shared_Best_Solution) < Makespan(Incumbent) Then
      Incumbent ← Shared_Best_Solution
Return Incumbent
```

The tabu search algorithm is a trajectory-based optimization method (Gendrau & Potvin, 2010); this approach aims to prevent the algorithm from getting stuck at local optima when exploring the solution space. In each iteration, the algorithm evaluates a set of possible moves, always selecting the best possible one (even if this implies *hill-climbing* move); in addition to this, the algorithm tries not to cycle (return to already visited solutions) during the search path, for this, it uses a list of forbidden moves (tabu).

Algorithm 2. Tabu_Search (Local search based on a Tabu Search scheme)

Input: *Shared_Best_Solution:* best solution found for all the asynchronous tabu algorithms*;*
Parameters: MAX_LOCAL: total number of local iterations, MAX_NEIGBORS: total number of neighbors randomly explored in the neighborhood;
Output: *Incumbent:* best scheduling found during the optimization algorithm

```
Current_Solution ← PerturbateSharedSolution(Shared_Best_Solution)
Best_Local_Solution ← Current_Solution
For i ← 1 to MAX_LOCAL
    changed ←False
    For j ← 1 to MAX_NEIGBORS
        Neighbor ← RandomSwap(Current_Solution)
        Neighbors.Add(Neighbor)
        If Makespan(Neighbor) < Makespan(Current_Solution) Then
            Current_Solution, changed ← Neighbor, True
        Else
            Neighbor ← RandomInsertion(Current_Solution)
            Neighbors.Add(Neighbor)
            If Makespan(Neighbor) < Makespan(Current_Solution) Then
                Current_Solution, changed ← Neighbor, True
            Else
                Neighbor ← RandomTwo-Opt(Current_Solution)
                Neighbors.Add(Neighbor)
                If Makespan(Neighbor) < Makespan(Current_Solution) Then
                    Current_Solution, changed ← Neighbor, True
        If Makespan(Current_Solution) < Makespan(Best_Local_Solution)
            Best_Local_Solution ← Current_Solution
    If changed ← False Then
        Current_Solution ← SelectBest(Neighbors)
Return Best_Local_Solution
```

Tabu search is handy in combinatorial optimization problems, such as scheduling or resource allocation, where the solution space is vast and cannot be evaluated exhaustively. Although tabu search does not guarantee the optimal solution, it can find high-quality solutions in a reasonable time and with moderate computational resources.

This work adapted the tabu algorithm to the flow shop problem using a vector-like encoding. The positions represent the order in which the jobs should be introduced to the machines, and the values contained in the vector explicitly encode the job identifiers. In addition, the tabu algorithm at each iteration sequentially evaluates a random neighbor of each movement (swap, insert, and 2-OPT); the search is configured to find a better-quality solution, and it updates the current solution (first-improvement). Finally, the algorithm manages to escape local optima since it allows for worsening. It is important to note that this proposed local search procedure does not use the concept of forbidden movement due to the low probability of cycling.

In this work, the team members have in common the use of the tabu search algorithm and share a centralized memory that allows them to access store, and update there, the best solution found. By encapsulating the algorithm in the members, the robustness of the architecture is acquired, replicating the methodology with variants facilitates the application of the technique, and diversity is generated with heterogeneous transition mechanisms that guide the members toward different solution spaces.

Algorithms 1 and 2 details the proposed asynchronous team of tabu search algorithms. It receives as input the total number of jobs to be scheduled, the total number of machines, and the processing time of each job on the different machines. It requires the correct tunning of four parameters. The result of the asynchronous team is the best sequence of jobs achieved during the optimization process.

4. Data Collection

The data for the application of the heuristic technique developed was obtained from the instances proposed by Taillard (1990), being a set of 90 instances distributed as shown in Table 1. All algorithms were programmed in Python 3.0, using the multiprocessing library. The experiments were run on a computer with an Intel ® Core ™ i9-12900K processor (Performance Cores: 8 Cores, 16 Threads, 3.2 GHz Base, 5.2 GHz Turbo and Efficient Cores: 8 Cores, 8 Threads, 2.4 GHz Base, 3.9 GHz Turbo) and 64 GB RAM.
The parameters of the AT_{TS} were set to MAX_GLOBAL = 15, MAX_TEAMS = 20, MAX_LOCAL=100, and MAX_NEIGBORS = the number of jobs of the instance problem. All the codes are open-source and are available in a repository (https://github.com/LuisTarazonaTorres/Asynchronous-team-for-flow-shop-scheduling-problem-a-case-study-.git).

Table 1. Distribution set of instances

Set	Instances	Jobs	Machines
20x5	10	20	5
20x10	10	20	10
20x20	10	20	20
50x5	10	50	5
50x10	10	50	10
50x20	10	50	20
100x5	10	100	5
100x10	10	100	10
100x20	10	100	20

5. Results and Discussion

5.1 Numerical Results

Table 2 shows the results of the best performing algorithms in the literature and the obtained results for the proposed AT_{TS}. The algorithms compared were: $NEGA_{VNS}$ proposed by Zobolas et al. (2009), HES_{SA} proposed by Khurshid et al. (2021), PSO_{ENT} proposed by Marinakis & Marinaki (2013), IG_RS y IG_RS_{LS} proposed by Ruiz & Stützle (2007) y RDO_{PSO} proposed by Chen et al. (2014). The obtained results for instances with five machines presented a percentage variation (GAP) of 0.00%, as the number of machines is increased to 10 and 20, the percentage variation increases and continues to increase if the number of jobs is increased. This is due to the level of combinations that are made during the application of the AT_{TS}.

Table 2. Comparison against the best performing algorithms in the literature.

Set	$NEGA_{VNS}$	HES_{SA}	PSO_{ENT}	IG_RS	IG_RS_{LS}	RDO_{PSO}	AT_{TS}
20x5	0,00%	0,00%	0,00%	0,04%	0,04%	0,00%	0,00%
20x10	0,00%	0,00%	0,07%	0,25%	0,06%	0,00%	0,17%
20x20	0,00%	0,00%	0,08%	0,21%	0,03%	0,03%	0,21%
50x5	0,00%	0,01%	0,02%	0,04%	0,00%	0,02%	0,00%
50x10	0,77%	0,43%	2,11%	1,06%	0,56%	1,31%	2,09%
50x20	0,95%	0,78%	3,81%	1,82%	0,94%	2,19%	3,21%
100x5	0,00%	0,00%	0,09%	0,05%	0,01%	0,00%	0,00%
100x10	0,08%	0,12%	1,26%	0,39%	0,20%	0,48%	0,88%
100x20	1,31%	0,94%	4,37%	2,04%	1,30%	3,00%	2,99%
Average	**0,34%**	**0,25%**	**1,31%**	**0,66%**	**0,35%**	**0,78%**	**1,06%**

5.2 Graphical Results

Figure 2 shows in a better way the comparison of the GAP with the best work done, the average GAP of the whole set of instances takes a value of 1.06%, an acceptable value compared to the other averages and for the case study, considering that currently the scheduling of material handling activities for the heap leaching process is done manually.

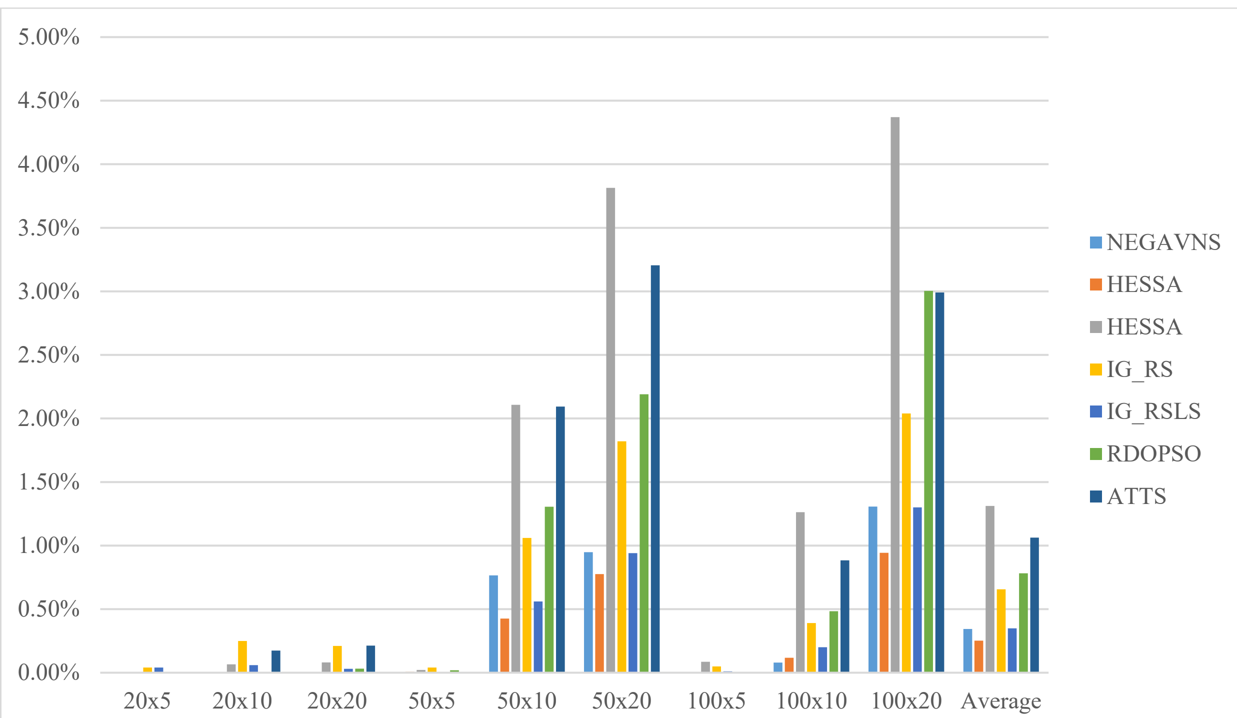

Figure 2. Benchmarking results for the 90 instances

5.3 Proposed Improvements

For instances with just five machines, results were obtained between 5 seconds to 180 seconds, these values are dependent on the number of jobs used in the application. For instances with a larger number of machines, results were obtained between 15 seconds to 600 seconds, also depending on the number of jobs. These values are acceptable for the case study, considering that the programming is performed manually in a time between 3 to 5 hours (and it cannot embed into the Digital Twin), thus, with the AT_{TS} it would be efficiently improving the obtaining of a result in a quite reasonable time. These results are good enough to embed the AT_{TS} into the Digital Twin.

5.4 Validation

The architecture consists of 24 threads (16 Performance Threads and 8 Efficient Threads). Figure 3 shows the level of thread usage, so that the AT_{TS} takes full advantage of the computational resources of the equipment that holds the Digital Twin. Furthermore, in the execution stage of the algorithm, there were no failures, corroborating the robustness that, in theory, the asynchronous team should have.

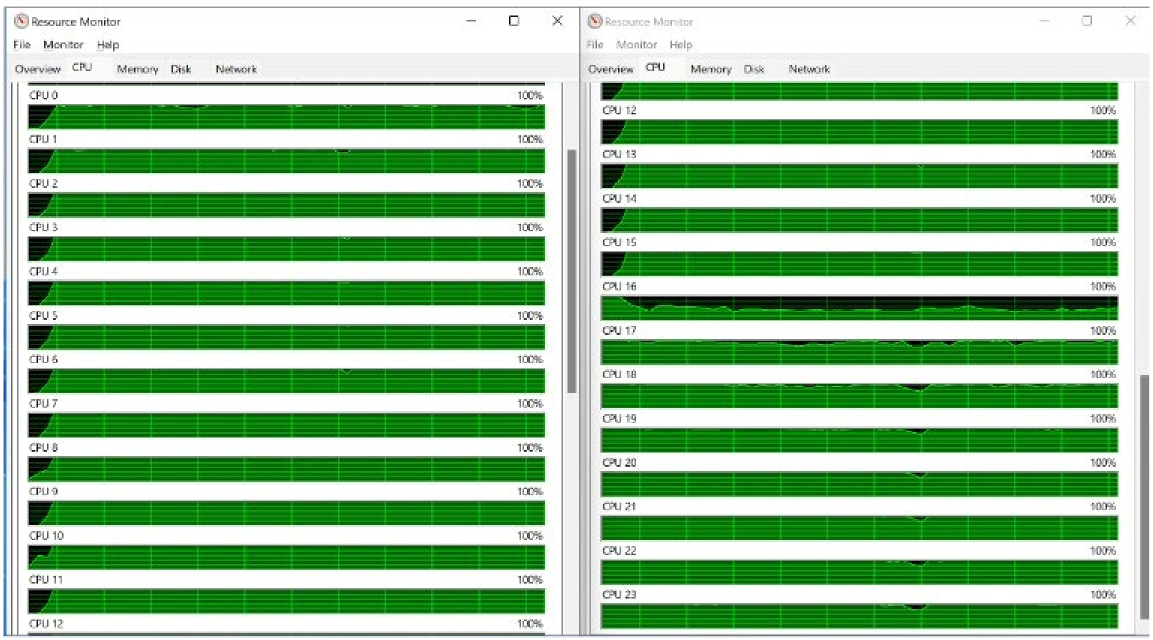

Figure 3. Utilization of 24 thread housed in the Digital Twin

6. Conclusion

A methodology using an asynchronous team architecture to solve the flow shop scheduling problem was developed for the case study of a Chilean mining company. The asynchronous team architecture of tabu search algorithms achieved the most optimums reported for classical instances in the literature, proving its excellent performance for cases with few machines.

The distribution of the asynchronous team across multiple processors allows us to exploit the computational architecture hosted by the Digital Twin. No singularities evidence the robustness of the asynchronous team in the execution.

The simultaneity and randomness of the tabu search algorithm offer the possibility to explore a much larger solution space, increasing the probabilities of the metaheuristic to find excellent quality solutions.

As future work, it is expected to contrast the improvements in the planning and control of material movement activities for the heap leaching process of the Chilean company with the asynchronous team integrated into the Digital Twin.

References

Abdel-Basset, M., Manogaran, G., El-Shahat, D., & Mirjalili, S. A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem. *Future generation computer systems*, vol. 85, no. 1, pp. 129-145, 2018.

Barrios, F. A. *Carguío dinámico para la optimización de la utilización efectiva de camiones de minería a cielo abierto a escala de laboratorio.* Santiago de Chile: Universidad de Chile, 2020.

Campbell, H. G., Dudek, R. A., & Smith, M. L. A heuristic algorithm for the n job, m machine sequencing problem. *Management science*, vol. 16, no. 10, B-630, 1970.

Checya, D. *Gestión de la operacion de equipos de movimiento de tierras para mejorar el rendimiento de carguío y acarreo en la mina antapacca.* Arequipa: Universidad Nacional de San Agustín, 2015.

Chen, C. L., Huang, S. Y., Tzeng, Y. R., & & Chen, C. L. A revised discrete particle swarm optimization algorithm for permutation flow-shop scheduling problem. *Soft Computing*, vol. 18, pp. 2271-2282, 2014.

Crainic, T. G., & Toulouse, M. Parallel meta-heuristics. In M. Gendreau, Potvin, & Jean-Yves, *Handbook of Metaheuristics*, pp. 497-541, New Yor: Springer. 2010.

Dannenbring, D. G. An evaluation of flow shop sequencing heuristics. *Management science,* vol. 23, no. 11, pp. 1174-1182, 1977.

Engin, O., & Güçlü, A. A new hybrid ant colony optimization algorithm for solving the no-wait flow shop scheduling problems. *Applied Soft Computing*, pp. 166-176, 2018.

Eunike, A., Wang, K. J., & Chiu, J. H. Real-time resilient scheduling by digital twin technology in a flow-shop manufacturing system. *Procedia CIRP,* vol. 107, pp. 668-674. 2022.

Gendreau, M., Potvin, JY. Tabu Search. In: Gendreau, M., Potvin, JY. (eds) Handbook of Metaheuristics. International Series in Operations Research & Management Science, vol 146. Springer, Boston, MA, 2010.

Hazrathosseini, A., & Afrapoli, A. M. The advent of digital twins in surface mining: Its time has finally arrived. *Resources Policy*, vol. 80, pp. 103155, 2022.

Johnson, S. M. Optimal two-and three-stage production schedules with setup times included. *Naval research logistics quarterly*, vol. 1, no. 1, pp. 61-68, 1954.

Khurshid, B., Maqsood, S., Omair, M., Sarkar, B., Ahmad, I., & & Muhammad, K. An improved evolution strategy hybridization with simulated annealing for permutation flow shop scheduling problems. *IEEE Access*, vol. 9, pp. 94505-94522, 2021.

Linn, R., & Zhang, W. Hybrid flow shop scheduling: a survey. *Computers & industrial engineering*, vol. 37, no. 1-2, pp. 57-61, 2023.

Liu, Y. F., & Liu, S. A hybrid discrete artificial bee colony algorithm for permutation flowshop scheduling problem. *Applied Soft Computing*, vol. 13, no. 3, pp. 1459-1463, 2013.

Marinakis, Y., & Marinaki, M. Particle swarm optimization with expanding neighborhood topology for the permutation flowshop scheduling problem. *Soft Computing*, vol. 17, pp. 1159-1173, 2013.

Nawaz, M., Enscore, E. E., & Ham, I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, vol. 11, no. 1, pp. 91-95, 1983.

Pandolfi, D., Villagra, A., & Leguizamón, M. G. Hibridización con búsqueda local de un algoritmo de estimación de distribución para la resolución del problema de secuenciamiento de Flow Shop. *XV Congreso Argentino de Ciencias de la Computación*, vol. 8, no. 5, pp. 14-25, 2022.

Peña, F., Órdenes, J., Wilson, R., & Navarra, A. Discrete Event Simulation for Machine-Learning Enabled Mine Production Control with Application to Gold Processing. *Metals*, vol. 12, no. 2, pp. 225, 2022.

Pinedo, M. *Scheduling: Theory, Algorithms and Systems.* New York: Springer, 2016.

Ponnambalam, S. G., Aravindan, P., & Rajesh, S. V. A tabu search algorithm for job shop scheduling. *The International Journal of Advanced Manufacturing Technology*, vol. 16, no. 10, pp. 765-771, 2000.

Rizkya, I., Syahputri, K., Sari, R. M., Siregar, I., Tambunan, M. M., Tarigan, U., & Isnaini, N. Nawaz, Enscore, Ham (NEH) Algorithm to Minimization of Makespan in Furniture Company. *IOP Conference Series: Materials Science and Engineering*, vol. 505, no. 1, pp. 012077, 2019.

Suliman, S. M. A two-phase heuristic approach to the permutation flow-shop scheduling problem. *International Journal of production economics*, vol. 64, no. 1-3, pp. 143-152, 2000.

Taillard, E. Benchmarks for basic scheduling problems. *European journal of operational research*, vol. 64, no. 2, pp. 278-285, 1990.

Taillard, E. Parallel Taboo Search Techniques for the Job Shop Scheduling Problem. *ORSA Journal on Computing,* vol. 6, no. 2, pp. 108-117, 1994.

Talbi, E. G. *Metaheuristics: from desing to implementation.* John Willey % Sons. doi:http://dx.doi.org/10.1002/9780470496916, 2009.

Talbi, E. G., Hafidi, Z., & Geib, J. M. A parallel adaptive tabu search approach. *Parallel computing*, vol. 24, no. 14, pp. 2003-2019, 1997.

Tliba, K., Diallo, T. M., Penas, O., Ben Khalifa, R., Ben Yahia, N., & & Choley, J. Y. Digital twin-driven dynamic scheduling of a hybrid flow shop. *Journal of Intelligent Manufacturing*, pp.1-26, 2022.

Zobolas, G. I., Tarantilis, C. D., & Ioannou, G. Minimizing makespan in permutation flow shop scheduling problems using a hybrid metaheuristic algorithm. *Computers & Operations Research*, vol. 36, no. 4, pp. 1249-1267, 2009.

Biographies

Luis Enrique Tarazona-Torres is a Ph.D. Engineering student and master's in industrial engineering at Los Andes University (Colombia), Chemical Engineer at the National University of San Antonio Abad del Cusco (Peru). Teaching Assistant in the Department of Industrial Engineering at Los Andes University (Colombia). His research area is Production and Logistics, especially in industrial production planning and scheduling, the application of mathematical and metaheuristic models for scheduling optimization. He has extensive knowledge and interest in the scheduling of activities in the industry and its connection with the implementation of Industry 4.0 technology.

Ciro Alberto Amaya-Guio is Ph.D. Industrial Engineering. École Polytechnique de Montréal, Canada Dissertation: Mathematical and computational modeling of vehicle routing problems in road marking activities. M. Sc. Industrial Engineering at Los Andes University (Colombia). Dissertation: Use of neural networks in manufacturing. Systems Engineer. National University of Colombia (Colombia). Black Belt Six Sigma, Arizona State University - ASU (USA). Associate Professor of Industrial Engineering at Los Andes University (Colombia). Researcher and consultant in production and logistics systems with a special interest in planning, control, and quality assurance; storage and distribution logistics; urban logistics, and hospital logistics. His solution methods are supported by mathematical models, metaheuristics, and discrete event-based simulation. Author of several academic publications, external peer, evaluator, and referee of academic journals and specialized conferences. He is currently the Faculty Advisor of IISE chapter 988.

David Álvarez-Martínez is Ph.D. in Electrical Engineering (Automation Science), São Paulo State University - UNESP (Brazil), M.Sc. in Electrical Engineering (Computer Science), Technological University of Pereira - UTP (Colombia), B.Sc. in Systems and Computer Engineering, Technological University of Pereira - UTP (Colombia), Postdoctoral Fellow, Applied Optimization Systems Group, Polytechnic University of Valencia - UPV (Spain) and Black Belt Six Sigma, Arizona State University – ASU (USA). Assistant Professor of Industrial Engineering at Los Andes University (Colombia). Director of the research group in Production and Logistics (PyLO). His main area of scientific activity is Operations Research and Computer Science. Within operations research, he focuses on the modeling, solution, and application of problems related to supply chain management, manufacturing and production, logistics and transportation, as well as material cutting problems, personnel hiring, goods packaging, and vehicle routing, among others. From a technical point of view, David's research focuses on the development of specialized software, combining exact and approximate optimization methods and artificial intelligence techniques applications, for the control of cyber-physical systems to solve real industry problems. David also supports public organizations in the definition of optimization, artificial intelligence instruments, and tools for the prospective analysis and strategic planning of prioritized agricultural chains in the country.