applied sciences

MDPI

Article

Centralized Evolutionary Optimization and Decentralized Agent-Based Scheduling for Open-Pit Mine Dispatching

Gabriel Icarte-Ahumada [1,*], Manuel Goyo [1], Victor Godoy [1] and Otthein Herzog [2,3]

[1] Faculty of Engineering and Architecture, Arturo Prat University, Iquique 1110939, Chile; mgoyo@unap.cl (M.G.); vngodoyr@unap.cl (V.G.)
[2] Department of Mathematics/Informatics, University of Bremen, 28359 Bremen, Germany; otthein.herzog@uni-bremen.de
[3] College of Architecture and Urban Planning, Tongji University, Shanghai 200092, China
* Correspondence: gicarte@unap.cl

Abstract

Truck dispatching in open-pit mining is an important process with a direct impact on productivity, equipment utilization, and hauling cost. While this problem is often treated as a sequence of isolated allocation decisions, a scheduling perspective makes it possible to coordinate truck movements and resource interactions over time. This paper compares two different approaches for schedule generation in open-pit mine dispatching: a centralized Genetic Algorithm (GA) and a decentralized Multi-Agent System (MAS). Both approaches were assessed using simulated instances based on real operational data from a Chilean open-pit copper mine. The evaluation considered transported material, travel time, hourly productivity, schedule generation time, and stability over repeated runs. The results indicate that the GA achieved higher average values in the main operational indicators. In particular, the GA achieved improvements of up to 22.4% in transported material and up to 26.4% in hourly productivity, while also showing lower average travel times. However, the MAS showed a computational advantage, reducing schedule generation time by up to 21.6%. These findings apply to static offline schedule-generation conditions; the relative advantages of the GA and MAS may differ in dynamic environments requiring repeated rescheduling, online adaptation, or real-time responses to operational disruptions.

Keywords: genetic algorithm; multi-agent systems; scheduling problem; truck dispatching

1. Introduction

Material handling is one of the most important processes in open-pit mining operations. In this process, trucks haul the material extracted by shovels to different destinations within the mine. Ore is transported to crushers or stockpiles, whereas waste material is transported to waste dumps. Since this operating cycle is repeated continuously throughout the shift, its performance has a direct effect on productivity and operating cost. Indeed, material handling may account for up to 50% of the total operational cost in open-pit mines [1].

Efficient truck dispatching in open-pit mining remains a difficult problem because mining operations evolve in highly dynamic and uncertain environments. Equipment failures, weather variations, and changes in road conditions may alter equipment availability and operational performance, generating delays and reducing the efficiency of the hauling process [2]. Under these conditions, determining the most appropriate next destination for each truck becomes a complex decision-making task.

Academic Editor: Kamil Szewerda

Received: 29 April 2026
Revised: 1 June 2026
Accepted: 9 June 2026
Published: 31 July 2026

Current practice commonly relies on centralized decision-support systems based on operations research techniques, heuristic approaches, or simulation models [3]. Many of these solutions follow a multistage strategy [1], in which an initial guideline is generated and then combined with dispatching criteria to make decisions in real time. Although such approaches are widely adopted, they do not ensure efficient operation. In practice, trucks may accumulate in queues at shovels or crushers while other loading resources remain underutilized. These situations increase inefficiency, raise operational costs, and may prevent the mine from achieving its production goals. Previous studies have argued that existing approaches often fail to model equipment activities with sufficient precision [4] and frequently depend on estimated information instead of the actual and changing state of the system [5–8].

To address these limitations, truck dispatching can be formulated as a scheduling problem, in which haulage decisions are coordinated over time rather than treated as isolated assignment events. Within this perspective, different computational approaches can be employed to generate schedules. Genetic Algorithms provide a centralized optimization-based approach that explores candidate schedules through evolutionary operators in order to improve operational performance. In contrast, Multi-Agent Systems provide a decentralized framework in which trucks and shovels are represented as autonomous agents that interact locally to construct schedules through negotiation. These two approaches reflect different ways of addressing the same scheduling problem: one through global evolutionary search and the other through distributed coordination among operational entities.

This study compares a Genetic Algorithm and a Multi-Agent System for the generation of schedules in open-pit mine truck dispatching. The MAS is derived from previous agent-based dispatching research, but its use for schedule generation in mining fleets still requires further comparative assessment against centralized optimization methods under equivalent experimental conditions. For this reason, the MAS is compared with a GA under equivalent simulation conditions, allowing the analysis to focus on the differences between decentralized negotiation and centralized evolutionary search. The purpose is to analyze how a centralized evolutionary optimization approach and a decentralized agent-based coordination approach perform when addressing the same scheduling problem under mining operational conditions. Through this comparison, the paper seeks to identify differences in their ability to generate efficient schedules for truck–shovel operations, thereby providing insights into the advantages and limitations of both approaches for mine dispatching. The comparison is carried out through simulated scenarios based on actual data from a Chilean open-pit mine.

The remainder of this paper is organized as follows. Section 2 reviews the related literature on truck dispatching and scheduling in open-pit mining. Section 3 describes the two approaches considered in this study, namely the genetic algorithm and the multi-agent system for schedule generation, and the experimental setup. Section 4 presents the obtained results. Section 5 discusses the main findings and their implications. Finally, Section 6 concludes the paper and outlines directions for future research.

2. Related Work

2.1. Mine Dispatching as an Optimization and Scheduling Problem

Mine truck dispatching has been widely recognized as an important operational process in open-pit mining because haulage constitutes a major share of total operating costs and directly influences equipment utilization, queue formation, and production continuity [9,10]. The literature has therefore approached dispatching primarily as an optimization problem in which trucks must be assigned to shovels and unloading points while satisfying production and operational constraints [11]. Across the reviewed studies, this problem has been addressed through mathematical programming, heuristic methods, stochastic formulations, and simulation-based frameworks, all of them seeking to reduce truck waiting times and shovel idle times while sustaining production targets [10,11]. At the same time, the literature shows that dispatching decisions are usually embedded in broader fleet management architectures in which routing, allocation, and real-time control interact as parts of the same operational system rather than as isolated decisions [10].

A distinction in the literature is the difference between allocation-based dispatching and explicit schedule generation. In many studies, the system determines the next destination of a truck every time a dispatch request occurs, which leads to an allocation-oriented formulation [10]. By contrast, a smaller number of studies models the problem as a scheduling problem in which an ordered sequence of future truck operations is explicitly constructed and later revised if the environment changes [3]. This distinction is important because scheduling formulations represent temporal interdependence more directly and can better capture queue propagation, equipment conflicts, and rescheduling needs under disruptions [12]. The reviewed studies also indicate that realistic mine environments complicate dispatching because of heterogeneous fleets, changing operating conditions, and the need to coordinate both loaded and empty truck movements under dynamic production requirements [10,12].

2.2. Genetic Algorithms for Mine Dispatching and Scheduling

Genetic algorithms have been one of the most recurrent metaheuristic approaches for mine dispatching and scheduling problems. Early GA-based studies focused on fixed-route or cost-oriented formulations, showing that evolutionary search could optimize truck numbers and route assignments under transport and maintenance cost criteria [13]. More recent studies extended this line toward multi-objective and real-time settings by incorporating heterogeneous fleets, transportation costs, truck waiting times, and operational constraints [9,14]. Some studies include representative examples of this evolution, such as multi-objective evolutionary algorithms for dynamic truck dispatching in heterogeneous operations, improved genetic algorithms with adaptive crossover, mutation, penalty, and renovation procedures for real-time dispatching, and enhanced GAs with elite preservation strategies to improve convergence and solution quality [15]. Additional evidence of the flexibility of GA-based scheduling is found in specialized mine transport contexts, such as battery-swapping trucks, where adaptive genetic algorithms were used to reduce both transportation cost and waiting time under additional energy-related constraints [16].

Although the GA literature demonstrates strong optimization capability, it also reveals some limitations. Most GA-based approaches remain fundamentally centralized because they rely on a global encoding of the dispatching or scheduling problem and frequently require tailored repair operators, penalty functions, or specialized chromosome representations to maintain feasibility [9,14]. Their performance is therefore often linked to problem-specific modeling assumptions and parameter tuning rather than to a more general adaptive control logic [15]. Moreover, when the operational environment becomes highly dynamic, centralized evolutionary search may face difficulties in reacting quickly to local disturbances, especially when rescheduling must occur during execution rather than before operations begin [12]. Thus, while genetic algorithms are powerful for complex combinatorial search, the literature suggests that strong global optimization does not automatically imply superior responsiveness in dynamic mine dispatching environments [9,12].

Beyond conventional GA-based approaches, recent studies show that open-pit mine scheduling and related mining optimization problems can also be addressed through swarm intelligence, hybrid metaheuristics, and exact optimization methods. PSO has been applied to stochastic long-term open-pit mine production planning under geological uncertainty, while recent DPSO formulations show how particle-swarm logic can be adapted to discrete scheduling structures such as assignment sequences and task schedules [17,18]. Hybrid GA–PSO approaches have also been used in mining-related optimization, including water-resource allocation in the Pinshuo open-pit combined mining area [19]. In addition, recent hybrid intelligent optimizers based on PSO have been proposed for long-term open-pit production planning under carbon-reduction constraints [20]. Other recent methods, such as MBHGA and MA-HCAGA, combine genetic operators with matrix encodings, clustering, discrete swarm components, and multi-agent logic, making them conceptually relevant for complex multiobjective and network-based scheduling problems, although they have not yet been directly validated in open-pit truck dispatching [21,22]. Finally, exact approaches such as mixed-integer programming and branch-and-bound remain important benchmarks for strategic open-pit mine production scheduling, even though their application to large-scale realistic instances may be constrained by computational complexity [23].

2.3. Multi-Agent Systems for Mine Dispatching and Scheduling

Compared with the broader GA literature, multi-agent approaches are less numerous in the literature for truck dispatching, but they address an important limitation of centralized dispatching: the difficulty of reacting efficiently to local changes and environmental dynamics. In the literature on multi-agent systems for truck dispatching in open-pit mines, trucks, shovels, and unloading points are represented as autonomous agents that negotiate schedules through an improved Contract Net Protocol [24]. This allows dispatching to be formulated not merely as a centralized assignment problem, but as a decentralized coordination process among interacting entities that hold local information and operational objectives. In this approach, schedules are generated explicitly, and when major unforeseen events occur, agents cancel previous assignments and regenerate schedules from the disruption point onward, which gives the system an intrinsic rescheduling capability.

2.4. Research Gap and Motivation

The reviewed literature reveals a clear gap at the intersection of scheduling representation and control architecture. First, a substantial part of the mine dispatching literature still focuses on allocation or route selection rather than explicit schedule generation, even though scheduling formulations provide a richer representation of temporal dependencies and rescheduling needs [10,12]. Second, genetic algorithms have shown strong capability for solving multi-objective mine dispatching and scheduling problems, but these approaches are predominantly centralized and are often tailored to specific cost, waiting-time, or energy-related objective structures [9,15,16]. Third, decentralized approaches such as multi-agent systems appear promising for disruption-prone and dynamic environments, yet they remain comparatively underexplored and are seldom contrasted directly with evolutionary approaches under the same scheduling perspective.

Accordingly, the main motivation for this study is to compare two contrasting paradigms for mine schedule generation. Genetic algorithms represent a centralized metaheuristic approach with strong global search capability, whereas multi-agent systems represent a decentralized coordination approach based on negotiation and local decision making. A direct comparison between these paradigms is relevant not only for identifying differences in solution quality and computational effort, but also for clarifying which control logic is more suitable for realistic open-pit mining environments characterized by uncertainty, heterogeneity, and rescheduling requirements. This gap provides the conceptual basis for the present work and justifies the comparison between GA-based scheduling and MAS-based scheduling under the same mine dispatching context.

The MAS considered in this study builds on previous agent-based dispatching work, where decentralized negotiation was introduced for open-pit mine truck dispatching. However, the use of the MAS as a schedule generation mechanism for mining fleets still requires further validation, particularly when compared with centralized optimization approaches under equivalent experimental conditions. Therefore, the novelty of the present article does not lie in proposing the MAS architecture as an isolated contribution. Instead, the contribution lies in using this MAS as a decentralized scheduling approach and comparing it with a centralized GA under the same scheduling formulation, mine representation, experimental scenarios, and performance metrics. This controlled comparison provides evidence on how decentralized agent negotiation performs relative to centralized evolutionary search when both are applied to the same mine fleet scheduling problem.

The MAS used in this study is not introduced as an unvalidated architecture. It builds on previous work in which the agent-based dispatching approach was evaluated against mathematical programming and historical production data from open-pit mining operations, and the simulation environment was previously validated in that context [3,24,25]. Therefore, the present article does not aim to revalidate the MAS or the simulator but to compare its decentralized scheduling logic with a centralized GA under equivalent experimental conditions.

3. Materials and Methods

3.1. Problem Description

Truck dispatching in open-pit mining takes place within a continuous and repetitive transport process. Once a truck completes its unloading task at a crusher, stockpile, or waste dump, it must be assigned a new shovel in order to continue the production cycle. From an operational perspective, this cycle includes the empty trip toward a loading point, the loading stage, the loaded trip toward the dumping destination, and the unloading activity, after which the truck becomes available again for a subsequent assignment. Because this sequence is repeated throughout the shift, the quality of each assignment decision directly affects the overall performance of the haulage system. Figure 1 depicts the activities in the truck cycle.

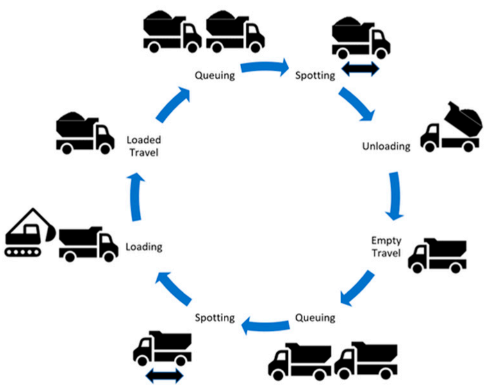

Figure 1. The truck cycle [3].

In most mining settings, these decisions are commonly handled by centralized dispatching systems. Such systems determine the next loading destination of each truck by relying on global information and allocation rules that typically consider indicators such as queue conditions, expected shovel productivity, or other predefined decision criteria [1,8]. Under this approach, dispatching is treated as a sequence of isolated assignment decisions made each time a truck finishes its previous cycle. As a result, the decision process tends to emphasize immediate system conditions rather than the longer-term consequences of repeated assignments.

However, the operating environment of an open-pit mine is inherently dynamic. Travel times may vary, road conditions may change, and unforeseen disruptions can alter the availability and synchronization of resources. Under these circumstances, a purely centralized and reactive allocation logic may generate undesirable effects in the system. For example, some shovels may accumulate excessive truck queues, while others remain underused, creating inefficiencies in both equipment utilization and material movement [7,26]. This motivates a scheduling formulation in which haulage decisions are coordinated over a planning horizon rather than treated as isolated assignment events. This perspective makes it possible to incorporate the relationships between successive decisions, capture temporal interactions among resources, and better anticipate congestion and coordination issues. In this sense, scheduling provides a broader representation of the operational process and offers a more suitable basis for improving resource utilization and material flow in complex mining environments [26–28].

3.2. Genetic Algorithm for Schedule Generation

3.2.1. Candidate Schedule Representation

In the proposed genetic algorithm, an individual I represents a complete candidate schedule for the truck fleet over the planning horizon, whereas the chromosome corresponds to the encoded form of that schedule. The chromosome is organized as a set of truck-specific decision sequences, so that each truck can be assigned to different shovels during the shift rather than being permanently associated with a single loading unit. Each gene represents one hauling cycle or dispatch decision and stores the loading and unloading choice assigned to a truck at a given decision stage. This representation preserves the dynamic reassignment logic required by the scheduling formulation while keeping the chromosome independent from explicit timestamps.

The individual I is represented as

$$I = \{G_1, G_2, \ldots, G_K\} \tag{1}$$

where K is the number of trucks, and G_k is the sequence of decisions associated with truck k:

$$G_k = \{g_{k,1}, g_{k,2}, \ldots, g_{k,r}\} \tag{2}$$

Each gene $g_{k,r}$ encodes the assignment selected for truck k in hauling cycle r and is expressed as

$$g_{k,r} = (s_{k,r}, u_{k,r}) \tag{3}$$

where $s_{k,r}$ denotes the shovel assigned to truck k in cycle r, and $u_{k,r}$ denotes the corresponding unloading destination. The chromosome does not store explicit start or completion times. Instead, it stores the sequence of operational decisions, while start times, waiting times, travel times, and completion times are computed during decoding. This separation between the decision layer and the temporal evaluation layer keeps the representation compact and allows the algorithm to evaluate the operational consequences of each candidate schedule after the assignments have been generated. A conceptual representation of this chromosome structure is shown in Figure 2.

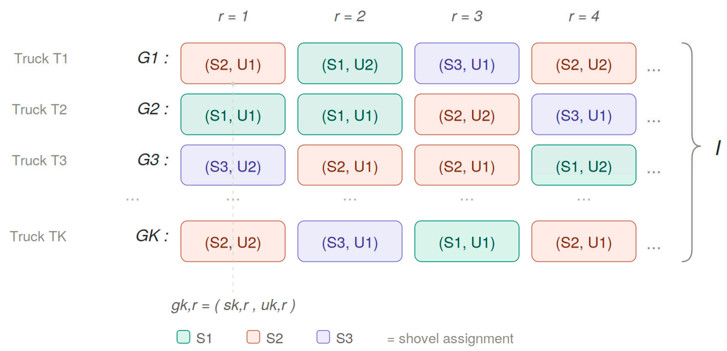

Figure 2. Conceptual chromosome structure for dynamic truck assignment.

3.2.2. Schedule Decoding and Feasibility Handling

Each chromosome is decoded into an executable schedule before its quality is evaluated. The decoding procedure simulates the sequence of activities implied by the genes of each truck, including the empty trip to the selected shovel, loading, loaded travel to the unloading destination, unloading, and the subsequent availability of the truck for its next assignment. Thus, the decoder transforms the assignment sequence encoded in I into a temporal schedule that can be assessed through production and travel-time indicators. This step is necessary because the chromosome itself only contains dispatch decisions and does not directly encode the timing of the operations.

For a gene $g_{k,r} = (s_{k,r}, u_{k,r})$, the decoder first determines the arrival time of truck k at shovel $s_{k,r}$ according to the completion time of its previous task and the empty travel time to the selected shovel. The loading start time is computed as the maximum between the truck arrival time and the availability time of the shovel. Once loading is completed, the loaded travel time to $u_{k,r}$ is added, and the unloading start time is determined according to the availability of the selected unloading resource. This procedure is repeated for all genes until the full schedule represented by I has been reconstructed.

Feasibility is checked during the decoding stage. A gene is feasible when both $s_{k,r}$ and $u_{k,r}$ belong to the admissible resource sets of the corresponding scenario. A decoded schedule is feasible when it respects truck availability, shovel availability, unloading capacity, and the shift horizon. When crossover or mutation generates an infeasible assignment, the resulting offspring is not evaluated as a valid schedule; instead, it is discarded and replaced by a newly generated feasible individual. This rule preserves the population size across generations and prevents infeasible schedules from influencing the fitness-based search.

3.2.3. Fitness Function and Normalization

The quality of each decoded schedule is evaluated through a fitness function that combines transported material and travel time. Let $M(I)$ denote the total transported material associated with individual I, and let $T(I)$ denote the total travel time accumulated by all trucks in the decoded schedule. Since the objective is to increase transported material while reducing travel time, both criteria are normalized and combined into a single scalar value. The fitness function is defined as

$$Fitness(I) = \lambda \bar{M}(I) + (1 - \lambda)(1 - \bar{T}(I)) \quad (4)$$

where $\bar{M}(I)$ is the normalized transported material, $\bar{T}(I)$ is the normalized travel time, and $\lambda \in [0, 1]$ controls the relative importance assigned to both terms. The transported material component is normalized within the evaluated population as follows:

$$\bar{M}(I) = \frac{M(I) - M_{min}}{M_{max} - M_{min}} \quad (5)$$

where M_{min} and M_{max} are the minimum and maximum transported material values observed in the current population. The travel-time component is normalized as

$$\bar{T}(I) = \frac{T(I) - T_{min}}{T_{max} - T_{min}} \tag{6}$$

where T_{min} and T_{max} are the minimum and maximum travel-time values observed in the current population. Since lower travel time is preferable, the term $(1 - \bar{T}(I))$ is used in the fitness function. Under this formulation, both components contribute to the same maximization direction: higher transported material increases the fitness value, and lower travel time also increases the fitness value. If all individuals in a population have the same value for one component, the corresponding normalized term is assigned the same value for all individuals to avoid division by zero.

It should be noted that population-level min–max normalization may exhibit reduced discriminatory power in early generations when the range of objective values is narrow. Although stochastic initialization without heuristic seeds promotes structural diversity from the first generation, no explicit diversity analysis was conducted in this study. Therefore, the effect of population diversity on the discriminatory capacity of the normalization procedure should be interpreted as a methodological consideration rather than as an empirically evaluated result.

3.2.4. Population Initialization and Evolutionary Cycle

The initial population is generated through a stochastic procedure. Each individual I is created by assigning truck decisions to admissible shovel–destination combinations defined in the corresponding scenario. No heuristic seed solution is incorporated during initialization. This design starts the search from a diverse set of feasible schedules and avoids introducing a dispatching rule that could bias the comparison with the MAS.

After initialization, each individual is decoded and evaluated using the fitness function defined above. The population is then sorted according to fitness. Parent individuals are selected using a fitness-ranking procedure, in which individuals with higher fitness have priority during reproduction. This selection mechanism gives better schedules a stronger influence on the next generation, while crossover and mutation maintain the exploration of alternative truck–shovel–destination combinations.

The evolutionary cycle consists of parent selection, crossover, mutation, feasibility verification, and replacement. After offspring are generated, each new individual is decoded and checked for feasibility before entering the population. Feasible offspring are evaluated using the same fitness function as the parent individuals. The cycle is repeated until the stopping criterion is reached. The final output of the GA is the best individual found during the search, together with its decoded schedule. The complete workflow of the genetic algorithm is summarized in Figure 3.

3.2.5. Genetic Operators and Replacement Strategy

The crossover operator combines information from two parent chromosomes to generate new candidate schedules. Since each individual I is structured as a collection of truck decision sequences G_k, crossover exchanges contiguous segments of hauling decisions between parent individuals. This allows an offspring to inherit productive subsequences from one parent and efficient assignment patterns from the other. In the context of dynamic truck assignment, crossover is relevant because good schedules may emerge from combinations of local decisions distributed across different truck sequences. A simplified example of the crossover operator is shown in Figure 4.

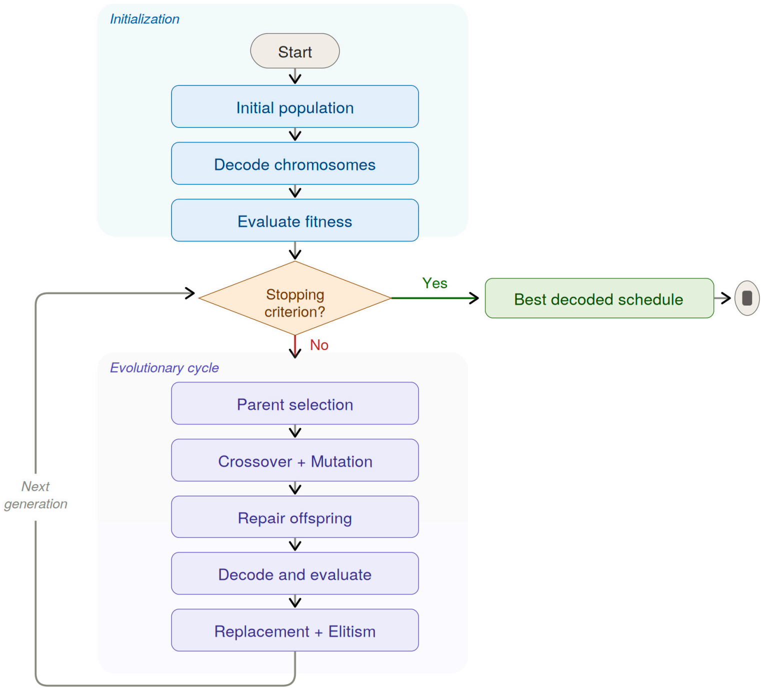

Figure 3. General procedure of the genetic algorithm.

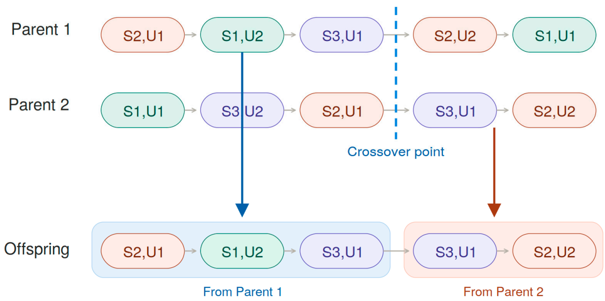

Figure 4. Example of crossover between two parent chromosomes.

The mutation operator introduces diversity by randomly modifying part of an individual. In this implementation, mutation may change the shovel $s_{k,r}$ or the unloading destination $u_{k,r}$ assigned in a gene, provided that the new value belongs to the admissible resource set of the scenario. Mutation may also perturb the order of decisions within a truck sequence when the resulting chromosome remains structurally valid. The purpose of mutation is to avoid premature loss of diversity and to explore dispatch alternatives that may not be generated through crossover alone.

The replacement stage is implemented as a population-retention strategy. After offspring generation, feasible parent and offspring individuals are ranked according to fitness. The best individuals are retained for the next generation according to the retention proportion defined in the experimental configuration, while the remaining positions are filled with feasible offspring. This mechanism should not be interpreted as classical elitism restricted to a small subset of individuals. Rather, it defines the survivor-selection rule used to maintain high-fitness schedules while still introducing new alternatives through crossover and mutation.

3.2.6. Parameter Configuration, Stopping Criterion, and Implementation

The GA parameters were kept fixed across all scenarios to ensure comparability with the MAS. The population size was set to 500 individuals, and the maximum number of generations was set to 10. The crossover rate was 0.4, and the mutation rate was 0.3. The fitness function used λ = 0.5, assigning equal weight to transported material and travel time after normalization. In the replacement stage, the best 80% of individuals were retained, while the remaining 20% of the population was replaced by feasible offspring generated through crossover and mutation.

The stopping criterion was the maximum number of generations. This value defined the computational budget used for the experimental comparison and should not be interpreted as evidence of convergence to a local or global optimum. Because the GA was limited to 10 generations, the stabilization of the best observed fitness value was not used as formal evidence of algorithmic convergence. The algorithm stopped after the predefined number of generations regardless of whether additional improvements might have been possible with a larger evolutionary budget. Therefore, the GA results reported in this study should be interpreted as the best schedules obtained under a fixed computational budget, rather than as solutions obtained after convergence.

No automatic hyperparameter optimization procedure was performed. Therefore, the parameter configuration should be interpreted as a fixed experimental setting rather than as an optimized GA configuration. The purpose of fixing these values was to compare the GA and MAS under the same scenarios and computational environment. Consequently, the reported results evaluate the behavior of this specific GA configuration and should not be interpreted as the best possible performance achievable by a genetic algorithm.

The GA was implemented in Java 1.8 using an object-oriented structure to represent individuals, populations, genetic operators, and the decoding procedure. The mine transportation network was represented using JGraphT version 1.0.1 as a weighted graph, where shovels, unloading points, and intersections corresponded to nodes, and haul roads corresponded to edges. This representation allowed the decoder to estimate route-dependent travel times and to generate schedules consistent with the spatial structure of the simulated mining environment.

3.3. Multi-Agent System for Schedule Generation

3.3.1. Agent Types and Local Information

The MAS is composed of three main categories of agents, each associated with a specific operational role within the mine: TruckAgents, ShovelAgents, and Unloading-PointAgents. These agents generate schedules through local interactions based on a modified Contract Net Protocol, which is described in Section 3.3.2. Each agent pursues an operational objective related to its role in the hauling cycle, so the final schedule emerges from the coordination of loading, hauling, and unloading decisions rather than from a centralized optimization process.

ShovelAgents represent the loading equipment and aim to generate schedules that maximize production. From the perspective of a shovelAgent, this means assigning trucks in a way that allows the shovel to load the greatest possible amount of material during the planning horizon. Its decision process considers shovel capacity, digging performance, loading rate, local availability, and the destination associated with the extracted material. Therefore, shovelAgents prioritize schedules that reduce idle time at loading resources and increase the continuity of the loading process.

TruckAgents represent haulage units and aim to generate schedules that minimize travel time while completing transportation tasks. Their decision process considers payload capacity, loaded and unloaded travel speeds, spotting time, unloading duration, current availability, and the mine road network. When evaluating alternative assignments, a truckAgent estimates the travel time required to reach a shovel, transport the material to the corresponding unloading destination, and continue with subsequent tasks. This allows each truckAgent to favor schedules that reduce unnecessary travel time and improve the use of haulage capacity.

UnloadingPointAgents represent material discharge facilities, such as crushers, stockpiles, and waste dumps. Their objective is to coordinate unloading availability in order to reduce waiting time and congestion at discharge points, thereby supporting a continuous material flow. These agents consider local service capacity, particularly the number of trucks that can be attended simultaneously, and the availability of unloading slots over time. By regulating access to discharge resources, UnloadingPointAgents help prevent excessive queues and support the feasibility of schedules that seek to increase production while reducing travel-related delays.

3.3.2. Adapted Negotiation Protocol and Concurrent Schedule Construction

The coordination mechanism is based on an adapted version of the Contract Net Protocol (CNP) [29], following the agent-based dispatching structure introduced in [3]. In this protocol, shovelAgents act as initiators and truckAgents act as potential contractors. The adaptation incorporates an explicit confirmation phase to handle concurrent negotiations. This is necessary because several shovelAgents may announce loading opportunities at the same time, and a truckAgent may participate in more than one negotiation before committing to a final assignment. The complete adapted negotiation protocol, including the confirmation phase used to handle concurrent negotiations, is shown in Figure 5.

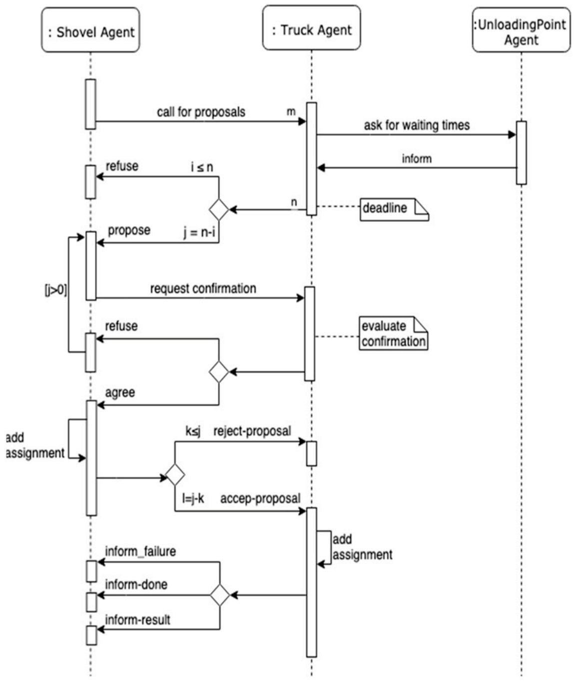

Figure 5. Adapted Contract Net Protocol with confirmation phase for handling concurrent negotiations.

A negotiation begins when a shovelAgent announces an available loading interval by sending a call for proposals (CFP) to the truckAgents. The CFP specifies the loading resource, the available time interval, and the unloading destination associated with the material to be transported. When a truckAgent receives a CFP from a shovelAgent, it evaluates whether the requested task can be inserted into its current schedule. This evaluation considers the truck's current availability, the travel time to the shovel, the loading operation, the loaded travel time to the unloading destination, the unloading operation, and the feasibility of the resulting schedule. If the task cannot be inserted without violating feasibility, the truckAgent sends a refusal. Otherwise, it submits a proposal containing its estimated arrival time and the operational times to accomplish the assignment.

The shovelAgent collects proposals until the negotiation deadline is reached or all contacted truckAgents have responded. If no feasible proposal is received, the negotiation closes without assignment. If feasible proposals are available, the shovelAgent ranks them according to the utility rule defined in Section 3.3.3 and sends a confirmation request to the truckAgent associated with the best-ranked proposal. Until this confirmation is accepted, the proposal remains tentative and does not modify the committed schedule of the truckAgent. The assignment becomes definitive only when the truckAgent accepts the confirmation request and inserts the corresponding decision into its schedule.

The confirmation phase resolves the concurrency problem at the truckAgent level. When a truckAgent receives a confirmation request, it checks whether the proposed assignment conflicts with other active negotiations. Let τ denote the confirmation threshold used by the protocol. In this implementation, $\tau = 60$ s. If the expected idle time of the requesting shovelAgent is greater than τ, the truckAgent accepts the confirmation to avoid a relevant idle period at the loading resource. If the expected idle time is less than or equal to τ, the truckAgent compares the active alternatives and accepts the assignment with the lowest travel time. The threshold $\tau = 60$ s is treated as a fixed protocol rule and is applied uniformly across all scenarios. It is not adjusted during the experiments and should not be interpreted as an optimized parameter. This value is inherited from the original MAS dispatching protocol and is used to resolve concurrent confirmation requests by defining when the reduction in shovel idle time takes priority over the selection of the alternative with the lowest truck travel time. Keeping τ constant avoids scenario-specific calibration of the MAS and preserves the comparability of both methods under fixed experimental conditions.

Ties among concurrent alternatives are resolved through a deterministic rule. The truckAgent first selects the alternative with the lowest travel time. If the tie persists, it selects the alternative with the earliest loading start time. If both values remain equal, the truckAgent accepts the confirmation request received first. Once an assignment is accepted, the other tentative proposals that conflict with the committed schedule are rejected or allowed to expire. If the selected truckAgent rejects the confirmation or does not respond within the allowed time window, the shovelAgent removes that proposal from the feasible proposal set, selects the next best-ranked proposal, and repeats the confirmation procedure. An example of a truck schedule generated through this protocol is shown in Figure 6.

Figure 6. Example of a truck schedule generated through the adapted negotiation protocol.

3.3.3. Proposal Evaluation and Utility Function

The proposal evaluation rule is defined to be consistent with the production–travel-time trade-off used by the GA. Both approaches evaluate schedules according to the same general objective: increasing transported material while reducing travel time. However, they apply this criterion at different decision levels. The GA evaluates complete candidate schedules through a centralized fitness function, whereas the MAS applies an equivalent criterion locally during the negotiation process.

Let $p_{s,k}$ denote a proposal sent by truckAgent k to shovelAgent s. Each proposal includes the estimated arrival time of the truck, the unloading destination associated with the material, the expected amount of transported material, and the estimated travel time required to execute the assignment. The travel time considers the empty movement toward the shovel and the loaded movement from the shovel to the unloading destination. Therefore, each proposal provides the information required to assess its local contribution to production and travel-time reduction.

The shovelAgent evaluates all feasible proposals using the same weighting logic adopted in the GA fitness function. The material and travel-time components are normalized within the set of feasible proposals received in the negotiation, and the same value of λ is used to balance both criteria. Under this rule, proposals that contribute more transported material and require lower travel time obtain higher utility values. This makes the MAS evaluation criterion compatible with the GA objective, although the MAS applies it locally rather than globally.

An important methodological distinction concerns the normalization reference sets used by both methods. In the GA, the normalized fitness components are computed with respect to the full population in each generation, which provides a global reference frame for comparing complete candidate schedules. In the MAS, the corresponding utility components are normalized only within the set of feasible proposals received by a shovelAgent during a single negotiation round, which provides a local reference frame. Therefore, although both methods use the same λ value and pursue the same general objective of increasing transported material while reducing travel time, their evaluation mechanisms may generate different preference behavior because they compare alternatives over different reference sets.

The shovelAgent selects the proposal with the highest utility value:

$$p_s^* = \arg\max_{p_{s,k} \in P_s} U(p_{s,k}) \tag{7}$$

If two or more proposals obtain the same utility value, the shovelAgent selects the proposal with the lower travel time. If the tie persists, the proposal with the earliest estimated arrival time is selected.

3.3.4. MAS Implementation

The MAS was implemented in the Java Agent DEvelopment Framework (JADE, version 4.6.0) [30]. JADE provides support for agent lifecycle management, behavior specification, message exchange, and interaction protocols. These features were used to implement shovelAgents, truckAgents, unloadingPointAgents, and their communication through the adapted CNP.

The MAS was developed in Java 1.8. As in the GA implementation, the mine transportation network was represented using JGraphT [31] as a weighted graph. Shovels, unloading points, and intersections were modeled as nodes, whereas haul roads were modeled as edges with associated travel costs. This graph-based representation allowed truckAgents to estimate route-dependent travel times when evaluating CFPs and generating proposals.

The main experimental difference between the methods lies in the schedule generation logic: the GA generates schedules through centralized evolutionary search, whereas the MAS generates schedules through decentralized negotiation among agents. However, this difference also implies distinct evaluation mechanisms. The GA uses fitness-based selection over complete candidate schedules and population-level normalization, while the MAS uses utility-based proposal evaluation within individual negotiation rounds and local proposal-level normalization. Thus, although both methods use the same general production–travel-time trade-off and the same λ value, their decision processes are not behaviorally identical.

3.4. Experimental Design

3.4.1. Scenarios

The experimental setup reproduces a surface mining operation in which material is hauled from loading areas to different discharge facilities by a fleet of large-capacity trucks. To represent the main elements of this process, the simulation includes trucks, shovels, crushers, stockpiles, and waste dumps. The evaluation was carried out using three scenarios of 12 h shifts derived from real operating data from an open-pit copper mine in Chile. These scenarios differ in size, but all of them consider heterogeneous equipment operating. The parameterization of the resources, including speed and capacity values for trucks and shovels, was defined from actual mine data and is summarized in Table 1, while the composition of each scenario is presented in Table 2.

Table 1. Property values for the simulations.

Equipment	Property	Unit	Min Value	Max Value
Trucks	Velocity loaded	km/h	20	25
	Velocity empty	km/h	40	55
	Capacity	t	300	370
	Spotting time	s	20	80
	Current load	t	0	370
Shovel	Capacity	t	35	80
	Load time	s	8	30
	Dig time	s	8	20
Crusher	Equipment discharging	number of trucks	1	1
Stockpile	Equipment discharging	number of trucks	1	20
Waste Dump	Equipment discharging	number of trucks	1	20

Table 2. Simulated scenarios.

Scenario	Number of Trucks	Number of Shovels
Small	13	5
Medium	43	10
Large	91	15

To improve reproducibility, the operational data were transformed into a simplified simulation instance composed of resource parameters, scenario composition, and a graph-based transportation network. The resource parameters include truck speeds in loaded and empty states, truck capacity, spotting time, shovel capacity, loading time, digging time, and unloading capacity. These values are reported as ranges in Table 1. The scenario composition is defined by the number of trucks and shovels in each instance, as reported in Table 2. The transportation network was represented as a weighted graph, where loading points, unloading points, and intersections correspond to nodes, and haul roads correspond to edges. Edge weights represent distances in kilometers used to estimate travel times during schedule generation. The simulation environment and the MAS-based dispatching logic were previously validated using real production data and comparisons with mathematical programming models [3,24,25]. In the present study, this validated simulation basis was reused to ensure a controlled comparison between the GA and MAS schedule-generation mechanisms.

In each simulated shift, resource parameters were sampled within the ranges reported in Table 1 and kept fixed for the corresponding run. Loaded and empty travel times were computed from the graph-based route distance and the corresponding truck speed state. Loading and unloading operations were modeled as capacity-constrained activities: each shovel can serve trucks according to its local loading availability, while each unloading point can attend a limited number of trucks simultaneously according to the capacities reported in Table 1. All scenarios used the same 12 h planning horizon, the same resource-parameter ranges, and the same road-network representation for both the GA and MAS.

The operational data used to construct the scenarios were obtained from an open-pit copper mine in Chile. Due to confidentiality restrictions, the complete mine layout, exact coordinates, proprietary production records, and original simulation instances cannot be publicly disclosed. To support reproducibility within these restrictions, the article reports the non-confidential elements required to construct comparable experimental instances, including resource-parameter ranges, scenario sizes, resource capacities, the graph-based modeling approach, travel-time computation assumptions, unloading-capacity constraints, and the common 12 h planning horizon. The same data abstraction was used for both methods, ensuring that differences in performance are attributable to the schedule generation logic rather than to different simulation inputs.

3.4.2. Evaluation Metrics

The performance of the evaluated approaches was assessed using four quantitative metrics. Travel time was defined as the cumulative fleet travel time accumulated during the simulated shift, computed as the sum of empty and loaded travel times across all trucks. This metric was expressed in hours (h). Transported material was defined as the total quantity of material hauled during the simulation horizon and was expressed in tons (t).

In addition, hourly productivity was calculated as the ratio between the total transported material and the cumulative fleet travel time, providing an aggregated indicator of system efficiency. This metric was expressed in tons per hour (t/h). Finally, schedule generation time was used to quantify the computational effort associated with each run and was measured in minutes (min). Taken together, these indicators provide a consistent basis for evaluating both operational performance and computational demand.

4. Results

4.1. Stability Analysis Under a Controlled Computational Environment

All simulations were executed in a controlled computational environment to ensure that the obtained results depended exclusively on the behavior of the compared methods rather than on differences in hardware or software configurations. Specifically, all experiments were carried out on a 2023 MacBook Pro (Apple, Inc., Cupertino, CA, USA) equipped with an Apple M2 Pro chip and 16 GB of unified memory, running macOS Tahoe version 26.3.1. Under these controlled conditions, the stability of the methods was assessed through the standard deviation and the coefficient of variation (CV), based on 10 independent runs of each experimental configuration.

Table 3 shows that the GA was generally more stable than the MAS in the operational metrics directly related to schedule quality, namely transported material, travel time, and hourly productivity. For transported material, the GA exhibited lower CV values in the large and small scenarios, and both methods were very similar in the medium scenario. For travel time, the GA showed lower variability in all three scenarios, particularly in the small scenario, where the MAS presented a CV of 4.30% compared with 0.95% for the GA. A similar pattern was observed for hourly productivity where the GA again displayed lower relative variability.

Table 3. Stability analysis using coefficient of variation (CV, %).

Scenario	Method	Material CV (%)	Travel Time CV (%)	Hourly Productivity CV (%)	Generation Time CV (%)
Large	GA	0.33	0.21	0.35	14.97
Large	MAS	1.00	1.76	0.84	1.71
Medium	GA	0.60	0.60	0.19	8.88
Medium	MAS	0.63	0.91	0.65	2.82
Small	GA	0.66	0.95	1.56	9.72
Small	MAS	1.07	4.30	3.39	10.20

However, the opposite occurred for schedule generation time. In the large and medium scenarios, the MAS was clearly more stable, with substantially smaller CV values than GA. In the small scenario, both methods showed similar variability, although the MAS still maintained a slight computational consistency advantage in absolute terms.

Overall, these findings indicate that the GA was more stable in the metrics associated with operational performance, while the MAS was more stable in the metric associated with computation time. This reinforces the interpretation that the GA provides better and more consistent schedule quality, whereas the MAS offers faster and more predictable schedule generation.

4.2. Comparison of Transported Material

Figure 7 presents the transported material obtained by both methods across the three experimental scenarios. In all cases, the GA achieved higher average production than the MAS. In the large scenario, the GA reached an average of 349,724.80 tons, while the MAS obtained 317,891.00 tons. In the medium scenario, the averages were 180,648.80 tons and 161,519.60 tons, respectively. In the small scenario, the difference became even more pronounced, with the GA obtaining 57,111.80 tons and MAS 46,676.40 tons.

These results indicate that the GA consistently generated schedules associated with higher production levels. The relative advantage of the GA over the MAS was approximately 10.0% in the large scenario, 11.8% in the medium scenario, and 22.4% in the small scenario. Therefore, although both methods were able to produce feasible schedules in all scenarios, the GA showed superior performance in terms of total transported material.

4.3. Comparison of Travel Times

Figure 8 shows the total travel times for both methods. The GA showed slightly lower average travel times than the MAS in all scenarios. In the large scenario, the GA required 443.21 h on average, compared with 447.26 h for the MAS. In the medium scenario, the averages were 214.48 h and 219.14 h, respectively. In the small scenario, the GA obtained 66.70 h, whereas the MAS reached 68.99 h. However, the Mann–Whitney U test did not detect a statistically significant difference in travel times. Therefore, although the descriptive tendency favors the GA, this result should be interpreted cautiously and should not be considered statistically confirmed.

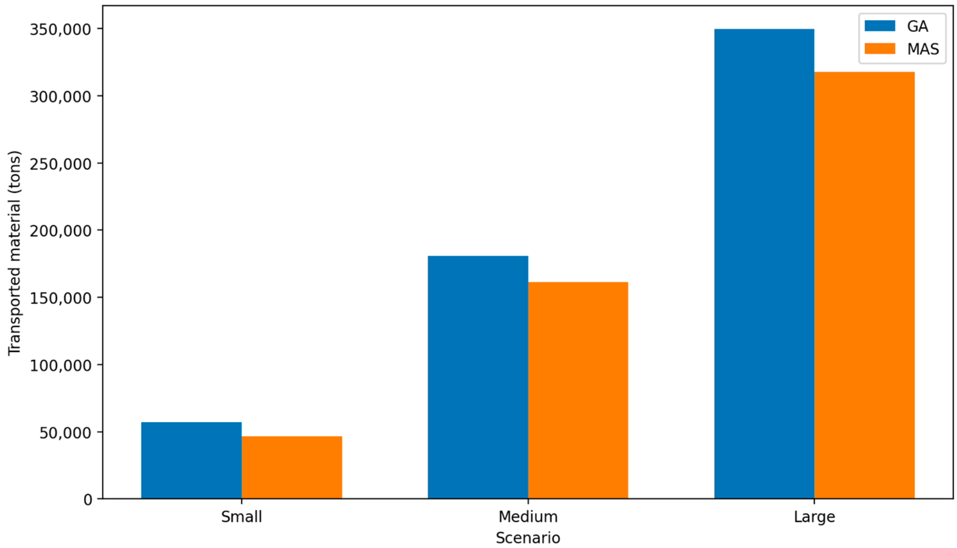

Figure 7. Mean transported material by scenario and method.

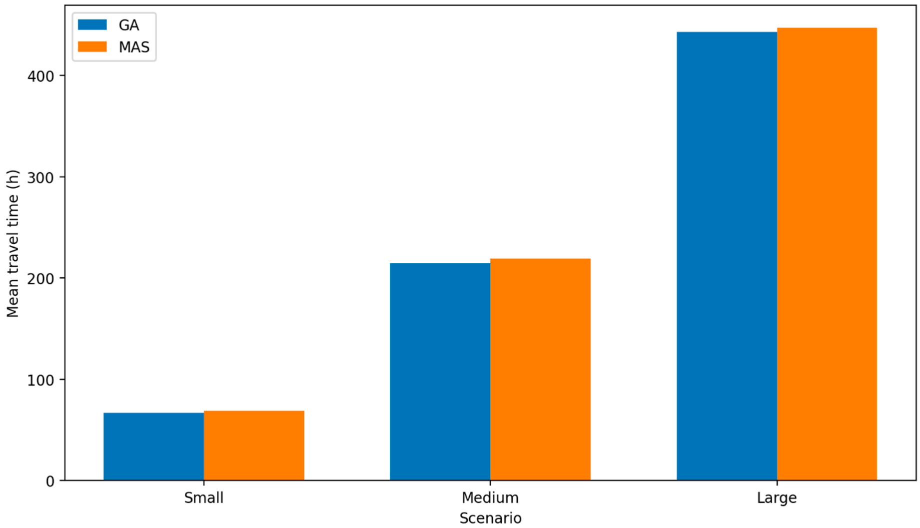

Figure 8. Mean travel time by scenario and method.

Although the absolute differences in travel time were smaller than those observed for transported material, the tendency was consistent across all scenarios. This means that the GA not only transported more material, but also did so with slightly less travel effort. Such a result suggests that the schedules generated by the GA were more compact and operationally better organized in terms of fleet movement.

4.4. Comparison of Hourly Productivity

Hourly productivity was computed as transported material divided by total travel time. As shown in Figure 9, the GA outperformed the MAS in all scenarios. In the large scenario, the GA reached an average hourly productivity of 789.08 tons per hour, while the MAS obtained 710.83 tons per hour. In the medium scenario, the averages were 842.22 tons per hour and 737.11 tons per hour, respectively. In the small scenario, the GA achieved 856.34 tons per hour, whereas the MAS reached 677.35 tons per hour.

https://doi.org/10.3390/app16157595

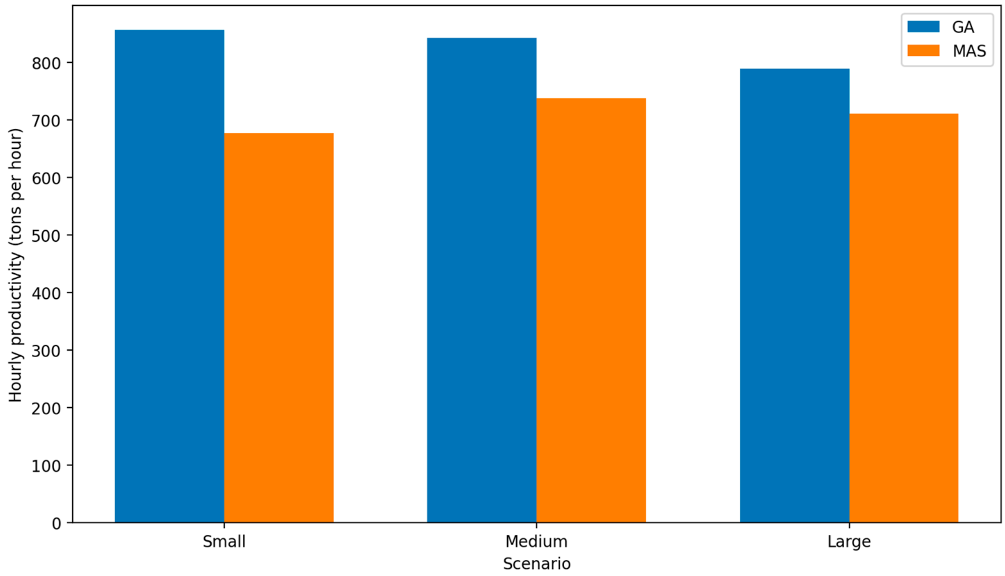

Figure 9. Mean hourly productivity by scenario and method.

This confirms that the advantage of the GA was not limited to absolute production. When production was normalized by travel effort, the GA still delivered better results. The relative improvement of the GA over the MAS was approximately 11.0% in the large scenario, 14.3% in the medium scenario, and 26.4% in the small scenario. Accordingly, the GA produced schedules that were more efficient from a production perspective.

4.5. Comparison of Schedule Generation Time

Figure 10 presents the time required by each method to generate the schedules. In the large scenario, the GA required 30.47 min on average, while the MAS required 23.88 min. In the medium scenario, the corresponding values were 4.47 min and 3.53 min, respectively. In the small scenario, both methods required less than one minute, with average values of 0.21 min for the GA and 0.18 min for the MAS. These results show that the MAS required lower average schedule generation time in descriptive terms across the three scenarios.

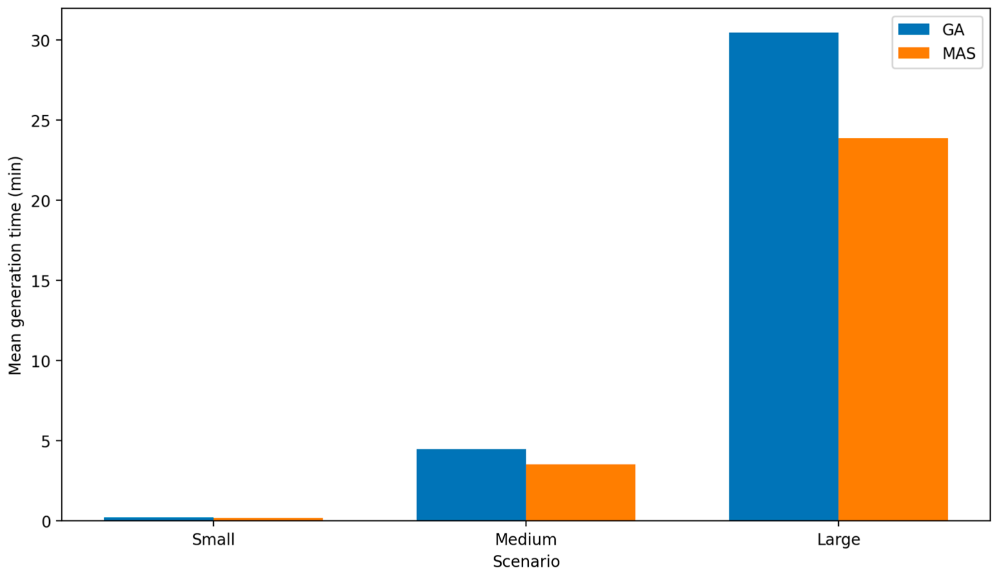

Figure 10. Mean schedule generation time by scenario and method.

4.6. Distributional Analysis of Results

To complement the mean-based comparisons presented in Figures 7–10, Figure 11 shows the distribution of the results obtained in the 10 independent runs for each performance metric, scenario, and method. This visualization allows the comparison to consider not only average values, but also the dispersion and consistency of the obtained results. The box plots confirm the general tendencies reported in the previous subsections: the GA achieved higher transported material and hourly productivity values, whereas the MAS required lower schedule generation times, particularly in the medium and large scenarios. Travel-time differences were less pronounced, especially in the large scenario, which is consistent with the inferential analysis reported in Section 4.7.

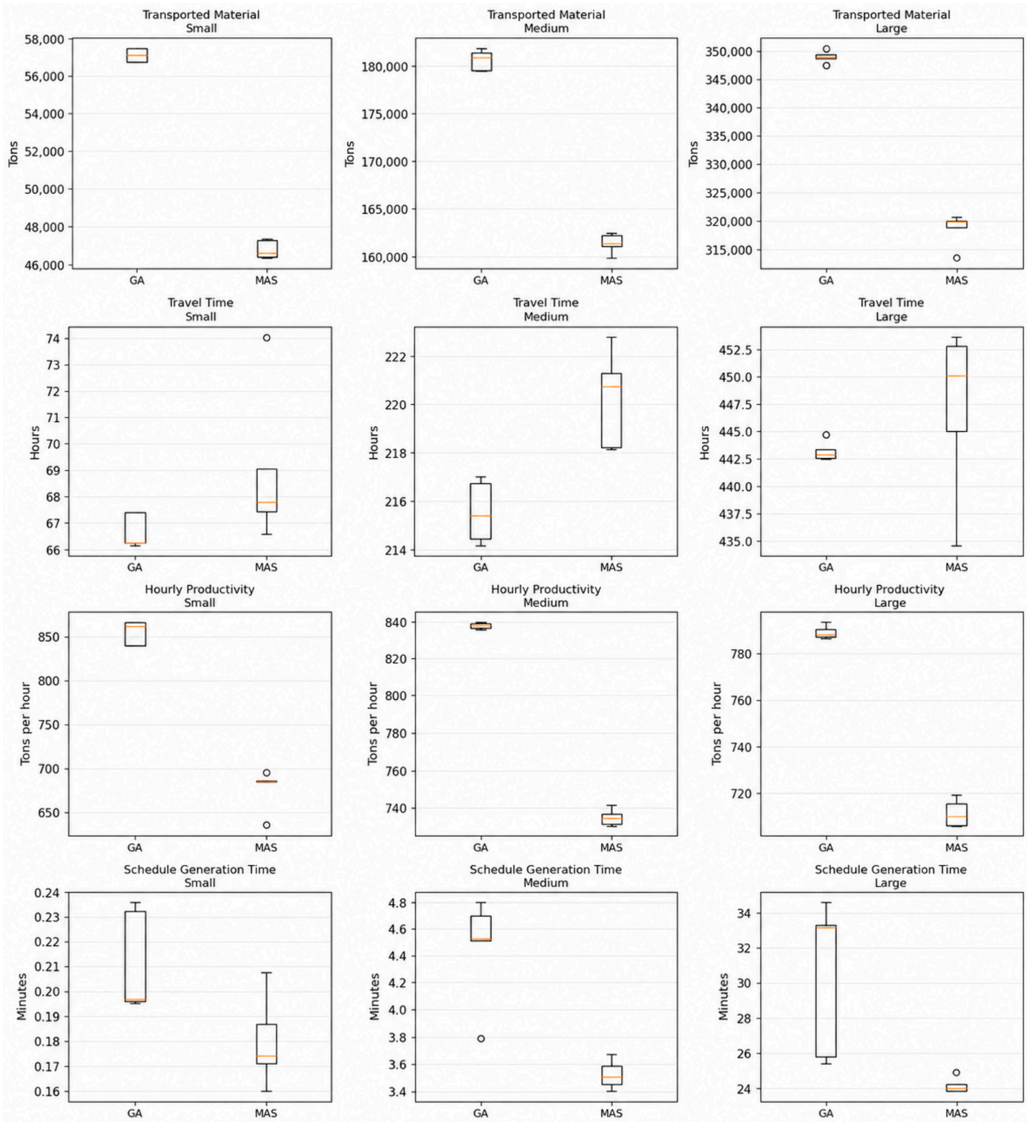

Figure 11. Box plot matrix of performance metrics by scenario and method.

4.7. Inferential Statistical Analysis

To complement the descriptive comparison, an inferential statistical analysis was performed using the Mann–Whitney U test. This non-parametric test was selected because of the reduced sample size and because it does not require assuming normality in the compared samples. The analysis was conducted independently for each scenario and metric, comparing the results obtained by the GA and MAS. The evaluated metrics were schedule generation time, material variability, travel time, and hourly productivity. A significance level of $\alpha = 0.05$ was used.

The results of the Mann–Whitney U test are reported in Table 4. In the large scenario, statistically significant differences were observed in schedule generation time, material variability, and hourly productivity, whereas travel time did not show a statistically significant difference. This indicates that, for the largest instance, both methods produced comparable travel-time results, although they differed in computational time and productivity-related indicators. Although the descriptive comparison shown in Figure 8 suggests a slight travel-time advantage for the GA, this difference was not statistically supported (p = 0.1508) and should therefore be interpreted as a descriptive tendency rather than evidence of a meaningful performance difference.

Table 4. Mann–Whitney U test results for GA and MAS comparisons.

Scenario	Metric	U-Statistic	p-Value	Significant Difference
Large	Schedule generation time	25.0	0.0079	Yes
Large	Material variability (CV)	25.0	0.0079	Yes
Large	Travel time	5.0	0.1508	No
Large	Hourly productivity	25.0	0.0079	Yes
Medium	Schedule generation time	25.0	0.0079	Yes
Medium	Material variability (CV)	25.0	0.0079	Yes
Medium	Travel time	0.0	0.0079	Yes
Medium	Hourly productivity	25.0	0.0079	Yes
Small	Schedule generation time	22.0	0.0556	No
Small	Material variability (CV)	25.0	0.0117	Yes
Small	Travel time	2.0	0.0356	Yes
Small	Hourly productivity	25.0	0.0117	Yes

In the medium scenario, all evaluated metrics showed statistically significant differences between the GA and MAS. The GA obtained better operational indicators in terms of travel time and hourly productivity, whereas the MAS maintained lower schedule generation times. This result reinforces the trade-off observed in the descriptive analysis: the GA tends to improve schedule quality, while the MAS reduces computational effort.

In the small scenario, no statistically significant difference was detected in schedule generation time. Therefore, for reduced instances, the computational difference between both methods was not statistically supported. However, statistically significant differences were observed in material variability, travel time, and hourly productivity. These results suggest that, even when computational times become similar in small instances, differences in operational performance remain observable.

Overall, the inferential analysis supports the interpretation that the GA and MAS exhibit different performance profiles. The GA tends to provide stronger operational indicators, particularly in travel time and hourly productivity, while the MAS shows an advantage in schedule generation time in the large and medium scenarios. However, the absence of a statistically significant difference in travel time for the large scenario and in generation time for the small scenario indicates that the relative advantage of each method depends on both the metric and the scenario size.

5. Discussion

5.1. Centralized and Decentralized Schedule Generation

The comparison between the genetic algorithm and the multi-agent system reflects two fundamentally different philosophies for schedule generation in open-pit mining. The genetic algorithm represents a centralized optimization approach, in which the schedule is built from a global search over candidate solutions. By contrast, the multi-agent system produces schedules through decentralized interactions among agents, where decisions emerge from local negotiations rather than from a single global optimization mechanism.

From this perspective, the observed behavior is coherent with what would be expected from both paradigms. The centralized approach is better positioned to exploit a broader view of the system and to coordinate assignments in a way that improves the global structure of the schedule. This helps explain why its schedules tend to be more effective in terms of productive performance and use of travel time. The decentralized approach, on the other hand, privileges autonomy, local responsiveness, and distributed decision making. Such an approach can generate feasible schedules without relying on a central optimizer, but its local nature may limit its ability to consistently reach the same global quality level obtained by the centralized search.

The unloading-capacity structure of the simulated mine provides an additional explanation for the observed differences. In particular, the crusher can attend only one truck at a time, whereas stockpiles and waste dumps allow higher concurrent unloading capacity. This constraint was applied equally to both methods and therefore does not introduce different experimental conditions. However, it may interact differently with each scheduling logic. The GA evaluates complete candidate schedules from a global perspective and can coordinate truck assignments while considering downstream congestion effects. In contrast, the MAS constructs schedules through local negotiations, which may limit its ability to anticipate the cumulative impact of a shared unloading bottleneck. Thus, the crusher capacity constraint should be understood as part of the common operational setting, but also as a factor that helps explain the differences observed between centralized and decentralized schedule generation.

A particularly relevant result is observed in the small scenario, where the GA achieved both higher average hourly productivity and lower relative variability than the MAS. Specifically, the GA obtained 856.34 t/h compared with 677.35 t/h for the MAS, while its coefficient of variation was 1.56% compared with 3.39% for the MAS. This combination suggests that, in the evaluated small instance, the centralized approach did not only improve average productive performance, but also produced more consistent results across repeated runs. Therefore, the advantage of the GA in this scenario is reflected not only in the magnitude of hourly productivity, but also in the stability of that performance.

This does not imply that decentralized generation is inferior in absolute terms. Rather, it suggests that each method optimizes a different dimension of the scheduling problem. The genetic algorithm is oriented toward solution quality under a centralized search logic, whereas the multi-agent system is oriented toward distributed generation, negotiation, and operational flexibility.

5.2. Trade-Off Between Solution Quality and Computational Time

A central finding of the comparison is the existence of a clear trade-off between solution quality and computational time. The genetic algorithm tends to deliver stronger schedules from the standpoint of productive performance, which suggests that the additional computational effort required by the evolutionary search is translated into better coordination of truck movements and resource usage. In other words, spending more time in schedule construction allows the centralized method to explore and refine solutions that later perform better during execution.

The multi-agent system exhibits the opposite behavior. Its decentralized mechanism allows schedules to be generated more quickly, which is an attractive feature in dynamic operational environments. However, that speed advantage is accompanied by lower operational performance in the evaluated static scenarios, indicating that faster generation does not necessarily lead to better operational outcomes. This is a relevant point because, in real mining settings, the best scheduling method is not always the one that maximizes a single operational metric, but the one that offers the most suitable balance between computational responsiveness and execution performance.

The trade-off becomes especially meaningful when considering the nature of the operational horizon. If schedules are produced for settings where decision time is limited and frequent rescheduling is required, a faster decentralized method may be preferable even if it sacrifices some performance. Conversely, if the priority is to obtain a more efficient plan and there is enough computational time before execution, a centralized evolutionary approach becomes more attractive. Thus, the choice of method depends not only on algorithmic superiority, but also on the temporal constraints and decision rhythm of the mining operation.

Nevertheless, the trade-off between solution quality and computational time should be interpreted with caution. The inferential analysis showed that not all descriptive differences were statistically significant. In particular, the travel-time difference in the large scenario and the schedule generation-time difference in the small scenario were not statistically supported. Therefore, these cases should be understood as descriptive tendencies rather than as conclusive evidence of method superiority.

5.3. Implications for Real Mining Operations

For real mining operations, these findings suggest that the selection of a scheduling strategy should be aligned with the operational context. In relatively stable environments, where input conditions do not change abruptly and there is time to compute schedules before implementation, a centralized method such as the genetic algorithm appears more suitable. Its stronger schedule quality would likely translate into higher productive performance and more effective use of haulage resources.

In contrast, decentralized schedule generation may become particularly relevant in environments characterized by uncertainty, disturbances, and frequent local changes. However, this should be interpreted as a theoretically motivated implication rather than as an empirical result of the present study, because the experimental design did not include dynamic disruptions, equipment failures, or online rescheduling events. The potential suitability of the MAS in such contexts is based on its structural characteristics, including local negotiation, distributed decision making, and the ability of agents to coordinate without relying on a single centralized optimizer. These characteristics are consistent with prior agent-based dispatching literature, in which decentralized coordination has been associated with flexibility and responsiveness under changing operational conditions. Therefore, while the present results demonstrate the behavior of the MAS under static scheduling conditions, its applicability to dynamic environments should be understood as an expected advantage that requires separate empirical evaluation.

This also opens the possibility of hybrid approaches. The results suggest that neither paradigm should necessarily exclude the other. A promising direction for real operations would be to use a centralized optimizer to produce a strong initial schedule and then rely on decentralized agents to adjust or repair that schedule during execution. Such a hybrid architecture could combine the global quality of centralized optimization with the flexibility and reactivity of decentralized coordination. From a practical standpoint, this may be more realistic than adopting either paradigm in a purely isolated form.

5.4. Contextualization with Related Work

The results of this study can be better understood when contrasted with the way prior work has applied genetic algorithms and multi-agent systems to mine truck dispatching.

In the GA literature, evolutionary methods have been used predominantly to optimize allocation decisions: determining the next destination of each truck at the moment it becomes available, based on criteria such as production targets, waiting times, or energy consumption [9,15,16]. Under that formulation, the GA operates on individual assignment events rather than on a complete sequence of future operations, and the quality of the solution is evaluated in terms of those isolated decisions. The present study differs in that the GA is applied to schedule generation: a chromosome encodes the full sequence of hauling cycles for every truck in the fleet, and fitness is evaluated over the decoded temporal schedule. This distinction matters for the comparison, because the GA results reported here reflect the quality of globally coordinated multi-truck schedules, not the cumulative outcome of repeated reactive assignments.

A similar distinction applies to the MAS literature. Prior agent-based dispatching work introduced decentralized negotiation as a mechanism for coordinating truck assignments in real time [12,18]. The MAS evaluated in this study builds on that architecture but applies it to schedule generation under the same formulation used by the GA, allowing a direct comparison between the two paradigms on equivalent ground. To the best of the authors' knowledge, a controlled comparison between GA-based schedule generation and MAS-based schedule generation under the same scheduling formulation, mine representation, and performance metrics has not been reported in the prior literature on open-pit mine dispatching.

From this perspective, the trade-off observed between solution quality and computational speed is not a restatement of what prior work already established. It is a finding specific to the schedule generation context: even when both methods pursue the same objective over the same planning horizon, centralized evolutionary search and decentralized agent negotiation produce systematically different outcomes in terms of operational quality and generation time. This result provides a basis for choosing between the two paradigms in practical scheduling applications, a question that the prior dispatching literature, focused on reactive allocation rather than explicit schedule construction, had not directly addressed.

5.5. Limitations of the Study

This study has limitations that should be acknowledged. First, the comparison is restricted to two specific implementations of schedule generation, namely the proposed GA configuration and the MAS-based scheduling mechanism, rather than to the full range of possible centralized, decentralized, evolutionary, swarm-based, hybrid, or exact optimization approaches. In particular, the experiments do not include direct comparisons with Discrete Particle Swarm Optimization, hybrid evolutionary algorithms such as GA–PSO or other advanced genetic variants, or exact optimization techniques such as mixed-integer programming, branch-and-bound, or branch-and-cut. Therefore, the results should not be interpreted as evidence of the general superiority of the GA over these alternative methods. Instead, the findings demonstrate that, under the static offline conditions, scenarios, metrics, and fixed parameter configuration evaluated in this study, the tested GA implementation outperformed the evaluated MAS in the main operational indicators. Consequently, broader conclusions regarding the relative performance of the GA, DPSO, hybrid evolutionary algorithms, and exact optimization methods require additional comparative experiments under equivalent scheduling formulations and computational settings.

Second, the evaluation focuses on aggregate performance indicators derived from the generated schedules. While this is appropriate for comparing scheduling outcomes, it does not fully capture other dimensions that are highly relevant in real operations, such as robustness to disturbances, ease of rescheduling, communication overhead, and adaptability during execution. These aspects are particularly important for decentralized systems and may alter the interpretation of their practical value.

Third, the experiments are based on predefined scenarios, which necessarily simplify the variability of real mining environments. Although scenario-based evaluation is useful for controlled comparison, actual operations may involve additional complexities such as stochastic breakdowns, changing priorities, heterogeneous dispatch rules, and human supervisory interventions. As a result, the relative strengths of the two methods could vary under more dynamic and less controlled conditions.

Finally, the study compares schedule generation methods from an offline performance perspective. This means that the faster computational behavior of the decentralized approach is observed in schedule construction, but its full advantage may only become visible in settings where repeated rescheduling or online adaptation is required.

6. Conclusions

This study compared two alternative approaches for schedule generation in open-pit mining operations: a centralized genetic algorithm and a decentralized multi-agent system. The analysis considered production-related and computational indicators, including transported material, travel time, hourly productivity, schedule generation time, and result stability under different scenarios. Overall, the comparison showed that both methods offer valuable capabilities for mine scheduling, differing in how they balance operational quality and computational speed, as discussed in Section 5.2.

It is important to note that the findings reported in this study are limited to the static offline scheduling conditions evaluated in the experiments and do not include dynamic disruptions, repeated rescheduling, or online adaptation during execution. Based on the results and discussion, the genetic algorithm provided better operational performance under static scheduling conditions, generating schedules with higher transported material, lower travel times, and greater overall hourly productivity. These findings indicate that, in environments where the system state is known in advance and schedules can be produced offline, the tested GA configuration was more effective for obtaining high-quality solutions. In contrast, the multi-agent system showed an advantage in schedule generation time, which reflects the benefits of decentralized coordination through local interactions and negotiation. From an operational perspective, this suggests that the GA approach evaluated in this study appears more suitable for pre-operational planning under the static offline conditions considered. However, this recommendation should be interpreted in relation to the fixed GA configuration used in the experiments, particularly the limit of 10 generations, which was adopted to preserve comparability rather than to maximize GA performance. A larger evolutionary budget could modify the observed performance–time trade-off by improving solution quality while also increasing schedule generation time.

Future work should extend this comparison to dynamic and stochastic mining environments, where disturbances, uncertainty, and real-time changes may alter the relative advantages of both approaches. In addition, hybrid strategies that combine the global optimization capability of genetic algorithms with the flexibility of multi-agent systems could provide a more balanced solution for practical applications. Another relevant direction is the incorporation of learning mechanisms into the agents, so that decentralized scheduling can improve over time and become more competitive in large-scale and highly variable mining operations.

Author Contributions: Conceptualization, G.I.-A. and O.H.; methodology, G.I.-A.; software, G.I.-A.; and V.G.; validation, G.I.-A. and O.H.; formal analysis, M.G.; investigation, G.I.-A.; resources, G.I.-A.; data curation, G.I.-A. and V.G.; writing—original draft preparation, G.I.-A. and O.H.; writing—review

and editing, M.G. and O.H.; supervision, G.I.-A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original operational data used in this study are subject to confidentiality restrictions associated with the industrial data provider and cannot be publicly disclosed. The manuscript provides the non-confidential parameter ranges, scenario composition, modeling assumptions, and methodological details required to construct comparable experimental instances under the same scheduling formulation.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GA	Genetic Algorithm
MAS	Multi-Agent System
CNP	Contract Net Protocol

References

1. Alarie, S.; Gamache, M. Overview of Solution Strategies Used in Truck Dispatching Systems for Open Pit Mines. *Int. J. Surf. Min. Reclam. Environ.* **2002**, *16*, 59–76. [CrossRef]
2. Adams, K.K.; Bansah, K.K. Review of Operational Delays in Shovel-Truck System of Surface Mining Operations. In Proceedings of the 4th UMaT Biennial International Mining and Mineral Conference, Tarkwa, Ghana, 3–6 August 2016; pp. 60–65.
3. Icarte, G.; Rivero, E.; Herzog, O. An Agent-based System for Truck Dispatching in Open-pit Mines. In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence*; Rocha, A., Steels, L., van den Herik, J., Eds.; SciTePress: Valletta, Malta, 2020; Volume 1, pp. 73–81.
4. Patterson, S.R.; Kozan, E.; Hyland, P. Energy efficient scheduling of open-pit coal mine trucks. *Eur. J. Oper. Res.* **2017**, *262*, 759–770. [CrossRef]
5. Chang, Y.; Ren, H.; Wang, S. Modelling and optimizing an open-pit truck scheduling problem. *Discret. Dyn. Nat. Soc.* **2015**, *2015*, 745378. [CrossRef]
6. Da Costa, F.P.; Souza, M.J.F.; Pinto, L.R. Um modelo de programação matemática para alocação estática de caminhões visando ao atendimento de metas de produção e qualidade. *Rem. Rev. Esc. Minas* **2005**, *58*, 77–81. [CrossRef]
7. Krzyzanowska, J. The impact of mixed fleet hauling on mining operations at Venetia mine. *J. S. Afr. Inst. Min. Metall.* **2007**, *107*, 215–224.
8. Newman, A.M.; Rubio, E.; Caro, R.; Weintraub, A.; Eurek, K. A review of operations research in mine planning. *Interfaces* **2010**, *40*, 222–245. [CrossRef]
9. Alexandre, R.F.; Campelo, F.; Vasconcelos, J.A. Multi-objective evolutionary algorithms for the truck dispatch problem in open-pit mining operations. *Learn. Nonlinear Model.* **2019**, *17*, 53–66. [CrossRef]
10. Wang, X.; Dai, Q.; Bian, Y.; Xie, G.; Xu, B.; Yang, Z. Real-time truck dispatching in open-pit mines. *Int. J. Min. Reclam. Environ.* **2023**, *37*, 504–523. [CrossRef]
11. Zeng, W.; Baafi, E.Y.; Fan, H. A simulation model to study truck-allocation options. *J. S. Afr. Inst. Min. Metall.* **2022**, *122*, 165–174. [CrossRef] [PubMed]
12. Shi, Y.; Ma, X.; Zhang, P.; Lu, H. A dynamic truck scheduling framework for open-pit mine production under equipment fault uncertainty. *Expert Syst. Appl.* **2026**, *299*, 130254. [CrossRef]
13. He, M.X.; Wei, J.C.; Lu, X.M.; Huang, B.X. The genetic algorithm for truck dispatching problems in surface mine. *Inf. Technol. J.* **2010**, *9*, 710–714. [CrossRef]
14. Yuan, W.; Li, D.; Jiang, D.; Jia, Y.; Liu, Z.; Bian, W. Research on Real-Time Truck Dispatching Model in Open-pit Mine Based on Improved Genetic Algorithm. In Proceedings of the International Conference on Cyber-Physical Social Intelligence, ICCSI 2022, Nanjing, China, 18–21 November 2022; pp. 234–239. [CrossRef]
15. Li, F.; Shi, Z.; Ding, W.; Gan, Y. Intelligent Optimization Scheduling Strategy for Energy Consumption Reduction for Equipment in Open-Pit Mines Based on Enhanced Genetic Algorithm. *Energies* **2025**, *18*, 60. [CrossRef]
16. Xiao, Y.; Zhou, W.; Luan, B.; Yang, K.; Yang, Y. Truck Transportation Scheduling for a New Transport Mode of Battery-Swapping Trucks in Open-Pit Mines. *Appl. Sci.* **2024**, *14*, 10185. [CrossRef]

17. Gilani, S.O.; Sattarvand, J.; Hajihassani, M.; Abdullah, S.S. A stochastic particle swarm based model for long term production planning of open pit mines considering the geological uncertainty. *Resour. Policy* **2020**, *68*, 101738. [CrossRef]
18. Wang, X.; Yao, W. A Discrete Particle Swarm Optimization Algorithm for Dynamic Scheduling of Transmission Tasks. *Appl. Sci.* **2023**, *13*, 4353. [CrossRef]
19. Chen, P.; Yang, J.; Duan, S.; Xie, X. Optimization of water resources utilization by GA–PSO in the Pinshuo open pit combined mining area, China. *Environ. Earth Sci.* **2022**, *81*, 126. [CrossRef]
20. Li, N.; Liu, J.; Wang, L.; Dai, B.; Zhao, S.; Chang, J.; Ye, H.; Yan, D. A hybrid intelligent optimization algorithm for long-term production planning of open-pit mine considering carbon reduction plan. *Swarm Evol. Comput.* **2025**, *98*, 102078. [CrossRef]
21. Akopov, A.S. MBHGA: A Matrix-Based Hybrid Genetic Algorithm for Solving an Agent-Based Model of Controlled Trade Interactions. *IEEE Access* **2025**, *13*, 26843–26863. [CrossRef]
22. Akopov, A.S.; Beklaryan, L.A. Evolutionary Synthesis of High-Capacity Reconfigurable Multilayer Road Networks Using a Multiagent Hybrid Clustering-Assisted Genetic Algorithm. *IEEE Access* **2025**, *13*, 53448–53474. [CrossRef]
23. Letelier, O.R.; Espinoza, D.; Goycoolea, M.; Moreno, E.; Muñoz, G. Production scheduling for strategic open pit mine planning: A mixed-integer programming approach. *Oper. Res.* **2020**, *68*, 1425–1444. [CrossRef]
24. Icarte ahumada, G.; Herzog, O. Application of multiagent system and tabu search for truck dispatching in open-pit mines. In Proceedings of the ICAART 2021—13th International Conference on Agents and Artificial Intelligence, Online, 4–6 February 2021; Volume 1, pp. 160–172. [CrossRef]
25. Icarte, G.; Berrios, P.; Castillo, R.; Herzog, O. A Multiagent System for Truck Dispatching in Open-pit Mines. In *Dynamics in Logistics. LDIC 2020*; Lecture Notes in Logistics; Freitag, M., Haasis, H.D., Kotzab, H., Pannek, J., Eds.; Springer: Bremen, Germany; Cham, Switzerland, 2020; pp. 363–373. [CrossRef]
26. Ozdemir, B.; Kumral, M. Simulation-based optimization of truck-shovel material handling systems in multi-pit surface mines. *Simul. Model. Pract. Theory* **2019**, *95*, 36–48. [CrossRef]
27. Kazemi Ashtiani, M.; Moradi Afrapoli, A.; Doucette, J.; Askari-Nasab, H. A Stochastic Energy-Efficient Robust Simulation-Based Truck Dispatching Optimization for Simultaneous GHG Mitigation and Operational Excellence in Open-Pit Mines. *Simul. Model. Pract. Theory* **2025**, *138*, 103026. [CrossRef]
28. Anaraki, M.G.; Afrapoli, A.M. Sustainable open pit fleet management system: Integrating economic and environmental objectives into truck allocation. *Min. Technol. Trans. Inst. Min. Metall.* **2023**, *132*, 153–163. [CrossRef]
29. Smith, R.G. The Contract Net Protocol: High-level communication and control in a distributed problem solver. *IEEE Trans. Comput.* **1980**, *C–29*, 1104–1113. [CrossRef]
30. Bellifemine, F.; Caire, G.; Greenwood, D. *Developing Multi-Agent Systems with JADE*; John Wiley & Sons: Chichester, UK, 2007. [CrossRef]
31. Michail, D.; Kinable, J.; Naveh, B.; Sichi, J.V. JGraphT—A Java Library for Graph Data Structures and Algorithms. *ACM Trans. Math. Softw.* **2020**, *46*, 16. [CrossRef]

https://doi.org/10.3390/app16157595