International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Improving the Efficiency of Genetic Algorithms through Discrete-Event Simulation

Hoang Van Bay

1Faculty of Fundamental Technical, AD-AF Academy of Vietnam, Vietnam

A B S T R A C T

Genetic Algorithms (GAs) have been extensively utilized as powerful heuristic methods for addressing complex optimization problems across various scientific and engineering domains. Their ability to explore large and nonlinear solution spaces makes them suitable for solving problems where traditional methods fall short. However, a significant limitation arises when GAs are applied in dynamic environments, where system parameters and objectives are subject to frequent and unpredictable changes. In such contexts, the performance of GAs tends to degrade due to their sensitivity to real-time events, delays in adaptation, and a lack of mechanisms to respond promptly to environmental shifts.

To address this challenge, this paper introduces an integrated approach that leverages Discrete Event Simulation (DES) as a comprehensive framework for modeling, observing, and evaluating the performance of Genetic Algorithms in time-sensitive and non-static scenarios. Unlike conventional evaluation methods, which often assume static environments and fixed benchmarks, the proposed DES-based methodology enables the simulation of real-world processes through event-driven timelines and state transitions, providing a more nuanced and accurate representation of system behavior.

By embedding GA operations within a DES framework, the study enables real-time tracking of algorithmic convergence, robustness under varying conditions, and adaptability to dynamic constraints. This integration facilitates a deeper understanding of how GAs respond to changes over time, such as shifts in objective functions or system parameters. Extensive simulation results validate the effectiveness of this approach, demonstrating that DES-supported evaluation not only reveals performance bottlenecks but also uncovers opportunities for algorithmic tuning and structural enhancements.

The findings highlight the potential of Discrete Event Simulation as a valuable tool for advancing the development and deployment of Genetic Algorithms in real-world applications, particularly in domains characterized by uncertainty, variability, and temporal dependencies. This approach paves the way for more resilient, adaptive, and efficient optimization solutions in dynamic and complex systems.

Keywords: Genetic Algorithms, discrete Event Simulation, dynamic and complex systems.

Introduction

Optimization in dynamic systems remains one of the most critical and persistent challenges in the fields of operations research, logistics, manufacturing, and industrial automation. These systems are often characterized by high complexity, uncertainty, and temporal variability, where decision variables and constraints can change rapidly in response to real-time events such as equipment failures, shifts in demand, supply chain disruptions, or environmental fluctuations. In such scenarios, the ability to perform efficient and adaptive optimization becomes essential for maintaining performance, stability, and overall system efficiency.

Genetic Algorithms have emerged as a powerful class of evolutionary computation techniques capable of performing global search and optimization across complex, nonlinear, and multi-modal landscapes. Due to their population-based and probabilistic nature, GAs are inherently adaptable and have been successfully applied to a wide range of static optimization problems. However, when applied to dynamic environments, traditional implementations and evaluation methods reveal significant limitations. In particular, most GA studies rely heavily on static benchmark functions or artificially constructed test problems, which fail to capture the temporal dynamics, stochastic behavior, and event-driven changes found in real-world systems.

To overcome these limitations, researchers have increasingly sought more realistic frameworks for evaluating optimization algorithms under time-varying conditions. Among these, Discrete Event Simulation has proven to be a particularly promising approach. DES is a modeling technique that represents system behavior as a chronological sequence of discrete events, each of which alters the system state. Examples of such events include the arrival of a job, completion of a task, resource breakdown, or sudden changes in operational parameters. By incorporating temporal and causal relationships between events, DES allows for a much more accurate and detailed representation of complex systems, particularly those with asynchronous and time-dependent behaviors.

In this context, the integration of Discrete Event Simulation with Genetic Algorithms offers a novel and effective method for studying optimization in dynamic environments. This paper proposes a framework that embeds GA operations within a DES environment, thereby enabling real-time simulation of optimization processes under changing system conditions. Through this integration, we aim to evaluate the adaptability, convergence characteristics, and robustness of GAs when faced with dynamic constraints, variable objectives, and unpredictable event sequences. In doing so, the study seeks to provide a more comprehensive understanding of GA performance in realistic operational contexts and to identify strategies for enhancing algorithmic resilience in the face of environmental complexity.

Related Work

The integration of optimization algorithms with simulation techniques has been a subject of significant interest across multiple application domains, particularly in scheduling, routing, and resource allocation tasks. Genetic Algorithms have consistently demonstrated their value in solving such combinatorial optimization problems due to their flexibility, parallelism, and ability to escape local optima. Over the past few decades, numerous studies have applied GAs to real-world domains such as job-shop scheduling, transportation routing, warehouse operations, and energy management.

In parallel, Discrete Event Simulation has become a widely adopted modeling approach for capturing the time-dependent and event-driven behavior of complex systems. DES enables researchers and practitioners to analyze system dynamics by simulating sequences of events and tracking state transitions over time. It is particularly well-suited for systems in which operations do not proceed in fixed time steps but are instead triggered by specific occurrences such as task completions, resource releases, or failures.

Despite the complementary strengths of GAs and DES, relatively few studies have pursued their tight integration into a unified framework. Many existing works tend to utilize these techniques independently or in loosely coupled configurations. For example, optimization is often conducted using a GA first, followed by simulation with DES to validate or analyze solution quality—rather than allowing the simulation to actively influence the evolutionary process during runtime.

Several notable contributions illustrate the growing interest in combining simulation and evolutionary algorithms:

Smith et al. (2018) employed simulation-based optimization using GAs for hospital resource allocation. Their approach demonstrated how simulation could be used to evaluate the real-world applicability of GA-generated solutions, but it did not incorporate simulation feedback into the optimization loop.

Chen & Liu (2020) used DES to evaluate the stability of evolutionary algorithms in dynamic environments. Their work highlighted the impact of system variability on algorithmic robustness but treated DES solely as a post-analysis tool.

Hoang Van Bay et al. (2025) explored the use of Discrete Event Simulation to enhance the efficiency of the Ant Colony Optimization (ACO) algorithm. While promising, their integration still lacked real-time interaction between the simulation and the optimization engine.

These studies underline an important gap: in most cases, DES and GAs are employed as separate stages in the optimization workflow rather than as interconnected components within a continuous feedback mechanism. This decoupling limits the ability of the optimization process to adapt to evolving system states in real time and reduces the potential for discovering truly dynamic and resilient solutions.

In contrast, the present study proposes a tightly integrated DES-GA framework in which simulation and optimization processes interact dynamically throughout execution. This tight coupling allows for adaptive fitness evaluation, on-the-fly responses to system events, and more realistic modeling of complex operational environments. By embedding the GA within a simulation context, this approach provides richer insight into the algorithm's real-time behavior and enables performance tuning in a way that aligns more closely with practical, event-driven system requirements.

Methodology

Overview of the DES-GA Framework

The proposed DES-GA framework integrates a Discrete Event Simulation (DES) engine with a real-time, adaptive Genetic Algorithm (GA). This hybrid setup enables the GA to optimize solutions in response to dynamic changes within the simulated environment. Unlike traditional offline evaluations, each chromosome is tested and scored within an evolving simulation, providing context-sensitive feedback that reflects real-world conditions.

Discrete Event Simulation Engine

A Discrete Event Simulation engine is a computational model that advances the simulation clock only when specific events occur. These events represent discrete points in time at which the system state changes. Within the DES-GA framework, the DES engine simulates the behavior of each candidate route in a virtual environment where traffic jams, accidents, or road closures may delay travel between cities.

The DES engine maintains an event queue, processes events such as "entering a city" or "delayed segment", and updates the system state accordingly. This mechanism allows the evaluation of chromosomes not only based on distance but also on their performance under realistic and uncertain conditions, leading to more resilient and adaptable solutions.

The workflow of the Discrete Event Simulation engine is illustrated in Figure 1.

Fig.1- Workflow discrete event simulation engine

Job Arrival: Each chromosome in the genetic algorithm is treated as a job that enters the simulation. Upon arrival, the job is scheduled for evaluation based on its assigned route through the system. This simulates the initiation of a delivery or service task within a logistical network, triggering event tracking and performance monitoring from that point onward.

Machine Failure and Recovery: During route traversal, simulated events such as vehicle breakdowns or road segment failures may occur. These disruptions are modeled as machine failures in the DES, temporarily halting progress along specific edges of the tour. Corresponding recovery events are scheduled to resume operation after a predefined delay, allowing the simulation to capture the impact of real-world disturbances on overall route performance.

Inventory Update: At each visited city, the vehicle may be required to deliver or pick up items. These actions are treated as inventory update events, which dynamically adjust the onboard inventory levels. Violations of inventory constraints—such as exceeding capacity or running out of required items—are penalized and reflected in the fitness score of the solution. This encourages the evolution of routes that are not only efficient but also feasible in terms of inventory management.

Genetic algorithm integration and DES-GA integration loop

The Genetic Algorithm seeks to find optimal configurations, scheduling sequences, resource assignments that minimize a defined objective function.

Problem representation: a solution (chromosome) is represented as a permutation of city indices:

$$C = [c_1, c_2, ..., c_n]$$

Where:

$c_i \in \{1, ..., n\}$ and $c_i \neq c_j$ for $i \neq j$

Fitness Function: minimize total travel distance

$$f(C) = \sum d\{c_i, c_{i+1}\} + d\{c_n, c_1\}$$

Where:

$$d\{ij\} = sqrt((x_i - x_j)^2 + (y_i - y_j)^2)$$

Genetic operators: selection, order crossover and swap mutation

$$P(C_i) = \frac{1}{f(C_i)} : \sum \frac{1}{f(C_j)}$$

The GA DES algorithm is implemented as shown in Figure 2. The detailed steps are as follows:

Step 1: Generation Initial

GA generates a pool of candidate solutions.

Step 2: Simulate Chromosome in DES

A permutation of tasks, encoded as a list of job IDs or task sequences.

Step 3: Collect Performance Data

Refers to the process of gathering quantitative results from the **DES** after simulating each solution (chromosome) generated by the **GA.**

Step 4: Fitness Evaluation

Each chromosome is input to the DES, and the system simulates its execution. The total makespan, delays, or energy consumption is computed and returned as fitness.

Selection: tournament or roulette wheel selection based on fitness values.

Crossover: ordered crossover (OX) or position-based crossover is applied for task sequences.

Mutation: swapping or reordering parts of the chromosome to introduce diversity.

Step 5: Stopping Criteria Met

A decision point where the algorithm checks whether it should terminate the evolutionary process or continue with further generations.

Simulation and Results Achieved

Experimental Setup

To evaluate the effectiveness of the proposed integration between Genetic Algorithms (GA) and Discrete Event Simulation (DES), a series of simulation-based experiments were conducted within a controlled computational environment. The objective of these experiments was to observe the behavior, convergence patterns, and robustness of the GA under dynamically changing system conditions, as represented through discrete events.

Simulation Tool and Platform: All simulations and algorithmic operations were executed using MATLAB R2016a, which provides built-in support for evolutionary computation and offers sufficient flexibility for implementing event-driven simulations. MATLAB was selected due to its strong compatibility with both GA optimization routines and discrete-event modeling tools, as well as its capability to manage iterative processes efficiently.

GA Implementation Details: The Genetic Algorithm was implemented using MATLAB's Optimization Toolbox, with custom extensions to support dynamic fitness evaluation based on feedback from the DES engine. Rather than using a fixed, static fitness function, the evaluation metric was dynamically updated in real time to reflect ongoing changes in resource availability, system bottlenecks, and environmental variables. This enabled the GA to adapt responsively throughout its evolutionary process.

Hardware and Software Specifications: Experiments were performed on a standard desktop computing platform with the following configuration:

Processor: Intel Core i7-9700K CPU @ 3.60GHz

Memory: 16 GB RAM

Operating System: Microsoft Windows 10 (64-bit)

This configuration ensured that simulations could be executed within reasonable time frames without requiring high-performance computing infrastructure.

Parameters: Population size, Crossover rate, Mutation rate, Number of generations, DES time frame.

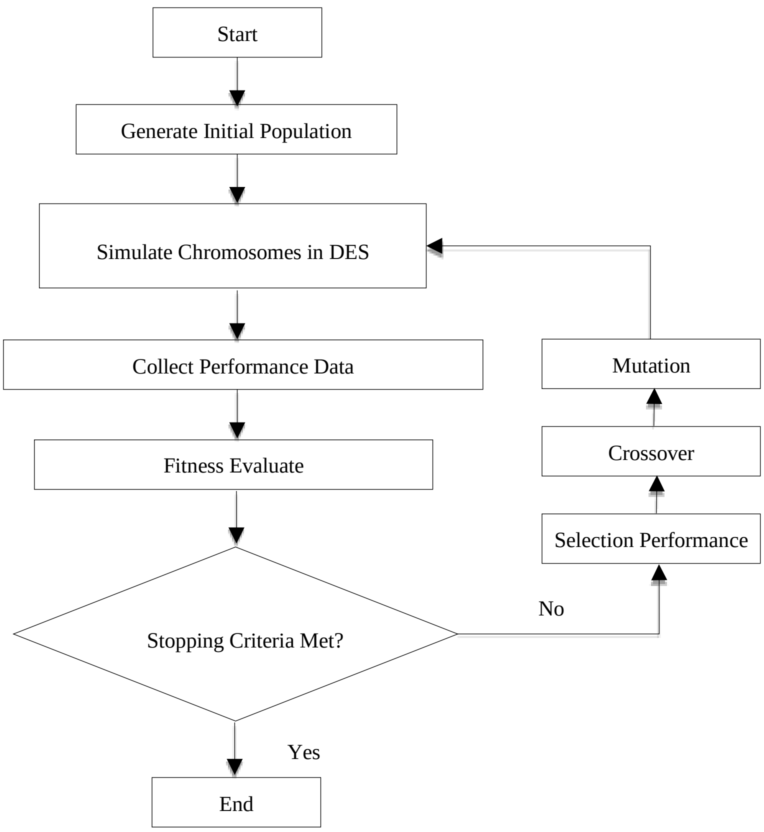


Fig.2- Block diagram DES-GA integration loop

Results

Table 1: Parameters setup

	popSize	nGen	crossProb	mutProb	numElite
1nd time	10	15	0.9	0.1	5
2nd time	20	30	0.9	0.2	7
3nd time	30	45	0.9	0.3	10

To implement the GA-DES algorithm, various simulation software tools can be utilized, such as Arena and MATLAB. In this study, the author employed the MATLAB programming environment (the source code is available for download from [21]) in conjunction with datasets from TSPLib, a widely used standard library containing sample instances of the Traveling Salesman Problem (TSP), including city coordinates and known optimal solutions.

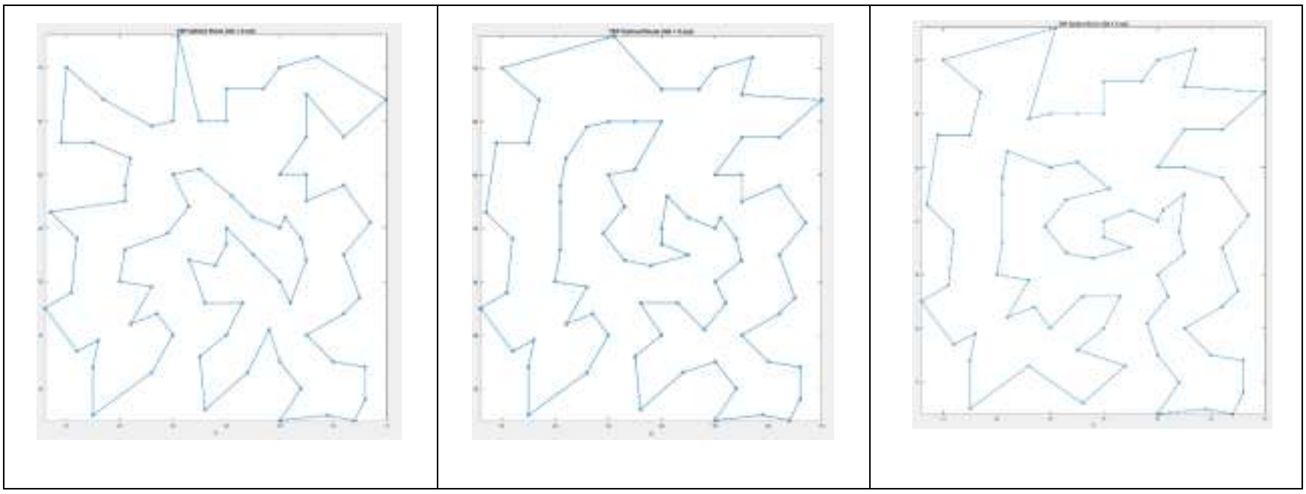
Two benchmark datasets from TSPLib were selected for experimentation: Eil76 and Berlin52. These datasets provide realistic problem instances of varying size and complexity, allowing for comprehensive performance evaluation. The simulation was executed using the hardware and GA parameter settings detailed in Table 1.

Table 2- Comparison of simulation results using the Eil76 and Berlin52 TSPLib datasets

	Name	Best	Achieved	Time(s)
1nd time	Eil76	538	571.3	10.4
	Berlin52	7542	7712.3	21.2
2nd time	Eil76	538	562.3	21.7
	Berlin52	7542	7613.4	28.5
3nd time	Eil76	538	551.3	25.3
	Berlin52	7542	7603.9	32.6

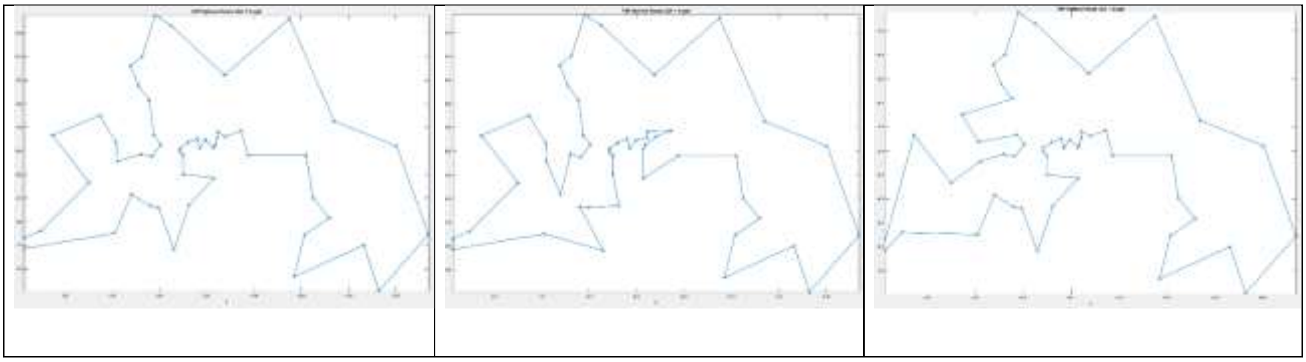
Table 2 presents the comparison of simulation results using the Eil76 and Berlin52 datasets. Each dataset was evaluated over three independent simulation trials, and the results are visualized in Figures 3 and 4, respectively. These figures depict the convergence behavior and adaptability of the GA under dynamically changing environmental conditions modeled by the DES engine.

The results indicate that the integrated GA-DES framework was capable of achieving near-optimal solutions while adapting to disruptions such as route delays, inventory constraints, and dynamic task arrivals. The variability between trials highlights the stochastic nature of both the GA and the event-driven environment, emphasizing the importance of repeated runs for statistical validity.



a) 1nd time **b) 2nd time** **c) 3nd time**

Fig. 3- Pathfinding results for the Eil76 dataset



a) 1nd time **b) 2nd time** **c) 3nd time**

Fig. 4- Pathfinding results for the Berlin52 dataset

Conclusion

The integration of Discrete Event Simulation with Genetic Algorithms represents a meaningful advancement in the domain of dynamic optimization. Traditional GA-based optimization methods often rely on static or overly simplified models, which fail to capture the complexity, uncertainty, and temporal dynamics inherent in real-world systems. In contrast, DES offers a detailed, time-aware modeling framework capable of simulating discrete state changes and event-driven behavior, thereby enabling a more realistic evaluation environment.

When tightly coupled with GA, this simulation-driven optimization approach transforms the evolutionary process from a static search over fixed fitness landscapes into a dynamic and responsive mechanism. Instead of relying solely on pre-defined objective functions, the GA is guided by real-time feedback from the simulated system, allowing it to adapt continuously to changes such as disruptions, fluctuating constraints, and emergent behaviors. This enables the generation of solutions that are not only optimal under nominal conditions but also robust and practically viable in dynamic environments.

One of the critical benefits of the DES-GA integration is its ability to model and evaluate edge cases and rare events—such as sudden failures, resource contention, or unexpected delays—that are typically ignored in static optimization frameworks. By embedding these events into the simulation, the evolutionary algorithm is naturally steered toward discovering solutions that maintain their effectiveness across a broad range of operational scenarios.

Experimental results presented in this study demonstrate that the DES-based GA framework exhibits improved convergence rates, greater robustness, and enhanced adaptability compared to traditional GA approaches. These advantages are particularly evident under conditions involving high variability and event-driven dynamics, as often encountered in logistics, manufacturing, and service systems.

Looking ahead, the proposed framework offers promising opportunities for application in smart manufacturing, adaptive scheduling, and supply chain logistics, where continuous system monitoring and rapid response to change are critical. Future research directions include real-time implementation, multi-agent coordination under DES-GA, and the incorporation of machine learning to further enhance adaptation and prediction capabilities.

In conclusion, the integration of Discrete Event Simulation with Genetic Algorithms marks a strategic shift toward intelligent, context-aware optimization methodologies. This approach bridges the gap between theoretical optimality and real-world practicality, enabling optimization solutions that are resilient, adaptive, and aligned with the dynamic nature of modern operational environments.

References

Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)

Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)

Law, A.M.: Simulation Modeling and Analysis, 5th edn. McGraw-Hill, New York (2014)

Gendreau, M., Potvin, J.Y. (eds.): Handbook of Metaheuristics. Springer, New York (2010)

Fishman, G.S.: Discrete-Event Simulation: Modeling, Programming, and Analysis. Springer, New York

Robinson, S.: Simulation: The Practice of Model Development and Use. Wiley, Chichester (2004)

Lin, S.: Computer solutions of the traveling salesman problem. Bell Syst. Tech. J. 44(10), 2245–2269 (1965)

Dorigo, M., Gambardella, L.M.: Ant colonies for the travelling salesman problem. Biosystems 43(2), 73–81 (1997)

Reimann, M., Doerner, K.F., Hartl, R.F.: D-ants: savings based ants divide and conquer the vehicle routing problem. Comput. Oper. Res. 31(4), 563–591 (2004)

Stützle, T.: ACO algorithms for the traveling salesman problem. Université Libre de Bruxelles, Belgium

Brucato, C.B.A.: The traveling salesman problem. B.A., Sonoma State University (2010); M.S., University of Pittsburgh (2013)

Chan, D.: IC insertion: an application of the travelling salesman problem. Int. J. Prod. Res. 27(10), 1837–1841 (1989)

Venkatesh, D.: Generation of genetic maps using the travelling salesman problem. Int. J. Sci. Eng. Res. 5(6) (2014)

Brezina, I.: Solving traveling salesmanship problem (TSP) using ant colony optimization. Int. J. Eng. Tech. Res., July (2018)

Priestley, H.A., Ward, M.P.: A multipurpose backtracking algorithm. Technical Report, January (2003)

Smith, A., et al.: Simulation-based GA for hospital resource optimization. J. Health Inform. (2018)

Chen, L., Liu, M.: DES in dynamic evolutionary optimization. Appl. Soft Comput. (2020)

Hoang, V.B., et al.: Application of discrete event simulation to enhance the efficiency of the ant colony optimization algorithm. ISSN 2582-7138 (2025)

TSPLIB benchmark datasets. http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/index.html

GitHub repository for GA-DES source code. https://github.com/vanbayhoang/GA_DES