






## Article

# Biased-Randomized Discrete-Event Heuristics for Dynamic Optimization with Time Dependencies and Synchronization

Juliana Castaneda <sup>1</sup>, Mattia Neroni <sup>2</sup>, Majsja Ammouriouva <sup>1</sup>, Javier Panadero <sup>3</sup> and Angel A. Juan <sup>4,\*</sup><sup>1</sup> Computer Science Department, Universitat Oberta de Catalunya, 08018 Barcelona, Spain<sup>2</sup> “Enzo Ferrari” Engineering Department, University of Modena and Reggio Emilia, 41121 Modena, Italy<sup>3</sup> Department of Management, Universitat Politècnica de Catalunya—BarcelonaTech, 08028 Barcelona, Spain<sup>4</sup> Department of Applied Statistics and Operations Research, Universitat Politècnica de València, 03801 Alcoy, Spain

\* Correspondence: ajuanj@upv.es

**Abstract:** Many real-life combinatorial optimization problems are subject to a high degree of dynamism, while, simultaneously, a certain level of synchronization among agents and events is required. Thus, for instance, in ride-sharing operations, the arrival of vehicles at pick-up points needs to be synchronized with the times at which users reach these locations so that waiting times do not represent an issue. Likewise, in warehouse logistics, the availability of automated guided vehicles at an entry point needs to be synchronized with the arrival of new items to be stored. In many cases, as operational decisions are made, a series of interdependent events are scheduled for the future, thus making the synchronization task one that traditional optimization methods cannot handle easily. On the contrary, discrete-event simulation allows for processing a complex list of scheduled events in a natural way, although the optimization component is missing here. This paper discusses a hybrid approach in which a heuristic is driven by a list of discrete events and then extended into a biased-randomized algorithm. As the paper discusses in detail, the proposed hybrid approach allows us to efficiently tackle optimization problems with complex synchronization issues.

**Keywords:** discrete-event heuristics; biased-randomization; dynamic optimization; synchronization



**Citation:** Castaneda, J.; Neroni, M.; Ammouriouva, M.; Panadero, J.; Juan, A.A. Biased-Randomized Discrete-Event Heuristics for Dynamic Optimization with Time Dependencies and Synchronization. *Algorithms* **2022**, *15*, 289. <https://doi.org/10.3390/a15080289>

Academic Editor: Frank Werner

Received: 25 July 2022

Accepted: 15 August 2022

Published: 16 August 2022

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

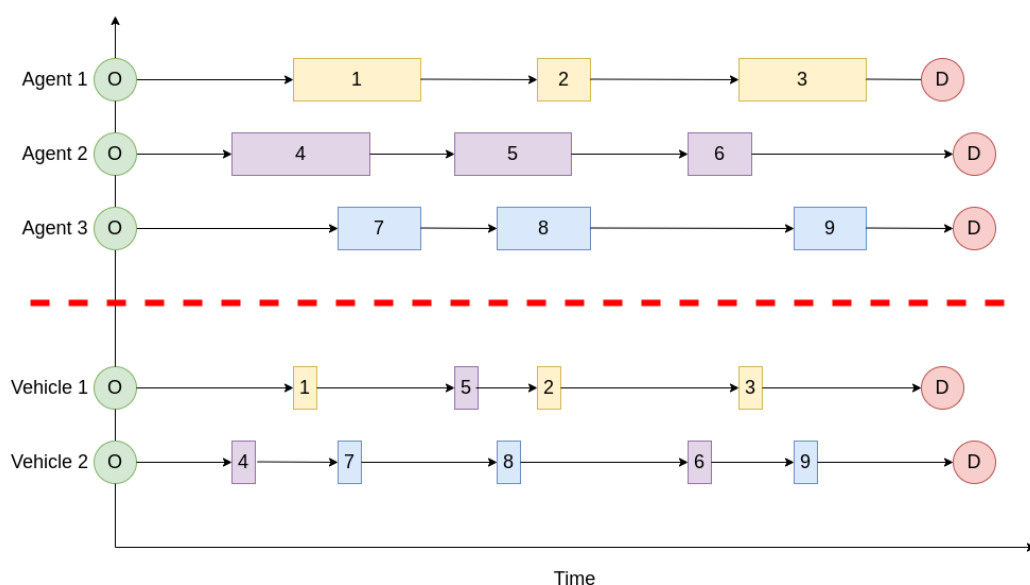


**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The industrial world today is characterized by much more complex systems and machines than in the past [1]. Companies are increasingly investing in integrating complex systems to boost their productivity and efficiency to gain an edge over their competitors. Hence, in modern companies, it is easy to see automated storage and retrieval systems (AV/RS) [2,3], automated guided vehicles (AGVs) [4,5], advanced software for transports optimization or production scheduling, and complex software architectures with a high level of parallelism. A similar process happened in cities, hospitals, airports and everyday services, where it is more and more common to see activities like car-sharing [6], bike-sharing [7], ride-sharing [8], short-term rental, and advanced management of personnel and flights [9]. These are all examples of real systems with a high degree of dynamism that require a good level of organization between different events and continuous process optimization. Figure 1 shows an illustrative example of a dynamic optimization problem with synchronization issues. A total of three agents, initially located at an origin depot (O), have to visit a series of locations (numbered from 1 to 9) and then return to a destination depot (D). In order to move around the system, these agents employ a ride-sharing service provided by two vehicles. The box size associated with each location corresponds to the time each agent or vehicle spends at that location. In the solution depicted in this figure, vehicle 1 drives agent 1 to location 1, then takes agent 2 to location 5, and then drives agent 1 to locations 2 and 3 before leaving this agent at D. Similarly, vehicle 2 will drive agent 2 to location 4, then agent 3 to locations 7 and 8, then agent 2 again to location 6,

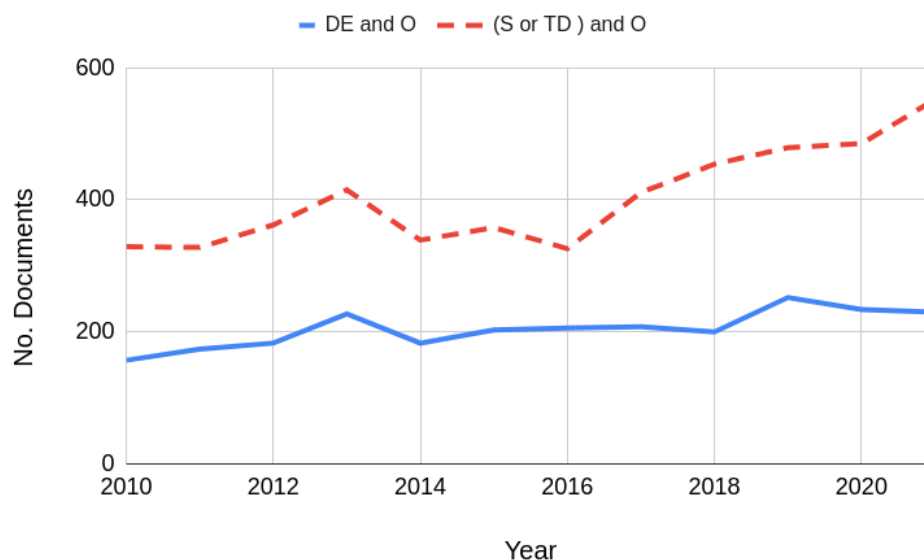
agent 3 to location 9, and finally agents 2 and 3 to D. Of course, in order for this solution to be feasible, we need to make sure that the vehicles can complete the assigned tasks on time (synchronization issue), which might depend upon which locations are assigned to each agent and which vehicle is assigned to each trip. Here, a goal such as finding a feasible solution that minimizes the total time invested in completing all the routing processes (makespan) could be considered, and many alternative solutions could be tested for feasibility and makespan value.



**Figure 1.** A simple example of a dynamic optimization problem with synchronization issues.

Due to the complexity of these systems, classical optimization approaches risk being ineffective or even detrimental. In fact, all classical optimization approaches (e.g., linear programming, dynamic programming, heuristics, metaheuristics, etc.) are based on the optimization of an objective function. If this objective function does not correctly represent the variables to be optimized in a system, the effort involved in the optimization model is almost useless. Following the seminal work of Fikar et al. [10], this paper describes a hybrid approach which combines concepts from discrete-event simulation (DES) [11] with heuristic algorithms [12] and biased-randomized techniques [13]. As we will discuss in this paper, the ‘discrete-event heuristic’ (DEH) approach allows us to efficiently deal with time dependencies and synchronization issues in dynamic optimization problems. The heuristic component allows the computation of effective solutions according to the problem constraints, while the DES enables the consistent estimation of the effect the provided solutions might have on the system. A DES is understood as a dynamic system that evolves by the occurrence of a series of discrete events [14]. The logic of discrete-event systems is found in multidisciplinary fields such as manufacturing systems, communication networks, traffic systems, computer systems, among others [15], and it could be used to represent both simple and complex processes, regardless of whether or not they present stochasticity. Hence, DEH constitutes a novel and powerful approach for solving dynamic optimization problems with time dependencies and synchronization issues [16].

Figure 2 shows the evolution, over the last decade, in the number of Scopus-indexed documents referred to the following searching terms: (i) ‘TITLE-ABS-KEY ((synchronization OR time dependencies) AND optimization)’; and (ii) ‘TITLE-ABS-KEY ((discrete event) AND optimization)’. One can notice that the first search seems to provide an increasing trend over the years, while the second one seems to be more stable in time. As discussed in this paper, however, there is a lack of effective methods in the literature to solve dynamic optimization problems with synchronization issues, which can be certainly complex due to the time dependencies they generate.



**Figure 2.** Number of articles indexed in Scopus for the main topics addressed in this article.

The main contributions of this work are: (i) to provide the theoretical fundamentals of the novel Biased-Randomized DEH (BR-DEH) methodology; (ii) to discuss some examples of how it can be used to solve dynamic optimization problems with time dependencies and synchronization issues; and (iii) to highlight future research perspectives related to BR-DEH algorithms. The remainder of this work is structured as follows. Section 2 provides a precise description of the concept of discrete event simulation. Next, a description of the fundamentals behind BR-DEH algorithms is presented in Section 3, and an overview of the optimization problems that can be solved with BR-DEH algorithms is offered in Section 5. Then, Sections 6 and 7 analyze existing works that employ similar approaches and summarize their computational results, respectively. Finally, conclusions and future research perspectives are reported in Section 8.

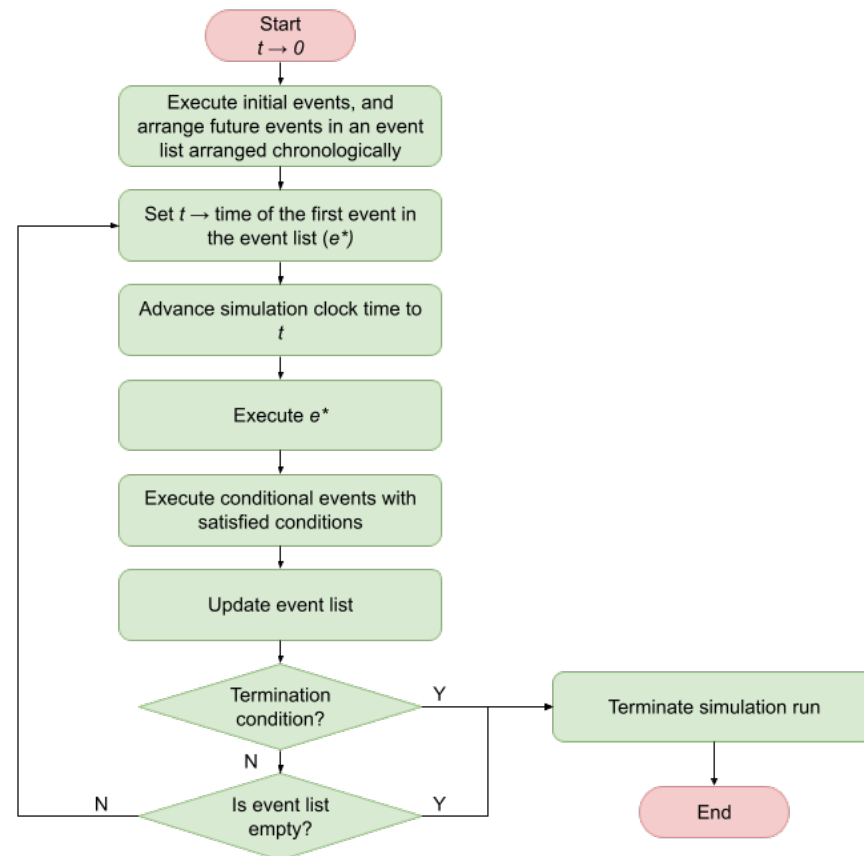
## 2. Discrete Event Simulation Concepts

Researchers benefit from DES concepts in studying systems [17–19]. The DES methodology enables the analysis of systems with time dependency between system elements. The name of DES refers to the use of events and the discrete-time advancement in the simulation execution. Events cause the state of a simulated model to change. For example, finishing the process of a job on a machine lets the machine become idle if no other job waits in its queue, and another machine could start processing the job in its manufacturing route. Another example is a postman arriving at a node in a network to deliver orders to the node. The time at which an event occurs defines the time advancement in the simulation model. Thus, a list of events that will occur in the future is defined and ordered chronologically according to events' time.

The events in DES are divided into events with a defined time to occur and are listed in the event list [18], e.g.,: the end of processing job  $i$  on machine  $A$ . This event time depends on the start of a process that lasts for a specific time, such as processing job  $i$  on machine  $A$ . Other events are conditional [19]. Their occurrence depends on satisfying one or more conditions. For example, processing job  $i$  on machine  $B$  requires an extra part to be delivered to the machine. If this part is not delivered, the processing of job  $i$  cannot start. Hence, the finish processing time cannot be scheduled yet.

At the execution of the DES, simulation clock time advances to the time of the first event in the event list, as illustrated in Figure 3. This event is executed, and changes are applied to the simulation model according to defined event routines. These changes include updating model variables and states. Therefore, new events are defined and added to the event list, and the executed event is removed from the list. This execution continues

until all events have been considered. Events with similar occurrence time are executed sequentially when the simulation clock time reaches their time. In addition, conditional events are checked at each time advancement to execute those with satisfied conditions.



**Figure 3.** Flow-chart illustrating the discrete event simulation running.

This execution of events enables the handling of dependencies between different elements in the model. As a result, complex systems are modeled and analyzed using the DES methodology. In addition, the defined relation between events helps to guarantee the synchronization of processes and elements in the simulated model. Most real-world systems are complex concerning the relation among system elements, such as production lines and logistics systems. Accordingly, the DES approach has become a candidate tool for analyzing these systems. For example, DES was used to analyze health care processes and support planning in them [20,21]. Ahalt et al. [22] evaluated different crowding scores in an emergency department using DES, and Demirli et al. [23] evaluated recommended solutions to reduce resource waste using DES. In transportation and logistics, Ref. [24] studied the impact of delivery time strategies on last-mile delivery distribution problems. In addition, DES can be used to analyze the stochastic behavior of systems because of the ability to generate observations from random variables during the execution of a simulation run. Researchers integrated DES into their developed frameworks to support decision-making, such as enhancing offshore service levels [25] and reducing material logistics costs in road construction [26].

### 3. Fundamentals of Discrete-Event Heuristics

The development of discrete-event based heuristics was motivated by the need to address realistic optimization problems in which synchronization of agents had to be considered. Thus, for instance, the movement of drivers has to be synchronized with the arrival of service users (riders) or objects that need to be transported. Thus, DEHs rely on the combined use of: (i) a discrete-event list that schedules the events happening in a

system (notice that new events might emerge in the future as a response to managerial decisions to handle past events, as in any discrete-event simulation process); and (ii) a sorted list of possible decisions associated with each occurring event, where these decisions are sorted by priority according to a specific efficiency criterion; and (iii) biased-randomized techniques, which allow us to quickly generate many alternative solutions, all of which are based on the efficiency criterion but introducing slight departures from them in order to explore alternative paths while handling the discrete-event list. Hence, in the context of house health care, Fikar et al. [10] analyze a realistic scenario in which an initial list of events is composed of nurses' and doctors' departures from a depot towards the houses they have to visit first. A fixed set of vans is employed to transport the medical staff using a ride-sharing mobility concept. Once the arrival events occur, new end-of-service events arise in the future, which generates the need to handle them by coordinating the vans so that nurses and doctors can be moved to their next visit. This process continues until no more visits need to be scheduled, and the health professionals can return to the depot. In parallel to this process, van drivers also have to be coordinated to pick up nurses and doctors after each house service. If the goal is to minimize the waiting times of nurses and doctors, the efficiency criterion should try to assign each member of the medical staff to the closest available van, so he or she does not have to wait for too long to be picked up and moved to his/her next destination. However, following a greedy criterion might clearly lead to sub-optimal solutions since short-term assignments might affect the efficiency of long-term ones. It is precisely here where biased-randomization strategies can offer a fast and effective way to explore solutions in the neighborhood of the greedy one so that randomness is introduced without losing the logic behind the efficiency criterion.

A schematic presentation of the simheuristic approach is shown in Figure 4. Given an optimization problem with synchronization issues or time dependencies among their potential tasks, we consider an iterative process that will help us to find a feasible solution of high quality. In each iteration, a list of discrete events is initialized. The events in this list are sorted according to their time of occurrence in the future, and they might be of different types (e.g., arrivals of a new order to a system, completion of a job processing, activation of a new service, etc.). Notice that, whenever an event is triggered, it might generate a new event that needs to be included in the list and scheduled into a future time (e.g., if a job starts being processed by a machine, and the processing time is known, a new event referring to the termination of this process will be included in the list). The generation of new events might depend on how previous events have been managed, so different ways of handling triggered events might lead to the generation of different events in the future (therefore, when managerial decisions are made during the processing of the events list, the actual configuration of this list might take one path or another depending on these decisions). The discrete-event list is being processed, always selecting the next event in time as the simulation clock advances. However, unlike a typical discrete-event simulation, every time a decision has to be made regarding how to handle this next event, an optimization component is called. This optimization component makes use of a heuristic following a logical criterion (e.g., if we need to assign a vehicle to a new service, the closest vehicle will be selected from the list of available vehicles; if we need to assign a task to a machine, from the list of available machines, the fastest one in completing that task will be selected, etc.). This mechanism allows us to incorporate a certain level of 'intelligence' into the simulation process. Still, in addition to that, our approach also adds a biased-randomization strategy, i.e., every time a decision has to be made to handle a new event, instead of acting in a greedy way (always selecting the vehicle, machine, etc., at the top of the operational-decisions list according to some logical criterion), we will assign decreasing probabilities of being selected to each of the available options for the decision maker. Hence, with a high probability, the element at the top of the operational-decision list will be selected, but sometimes (with a lower probability), the second one in the list will be selected, and sometimes (with an even lower probability), it will be the third one, etc. This way, we are introducing some 'oriented' (biased or non-uniform) randomization while processing

the list of events, which leads to a different configuration of this list, different values of the system status and, therefore, to a different solution in most cases. In summary, each time we start the initialization of the list of events, and thanks to this biased-randomized component, we are likely to obtain a new solution to our problem that, by design, will be feasible in terms of time dependencies (thanks to the discrete-event component) and of good quality—thanks to the logic behind the heuristic, which aims at prioritizing good decisions by assigning them a high probability.

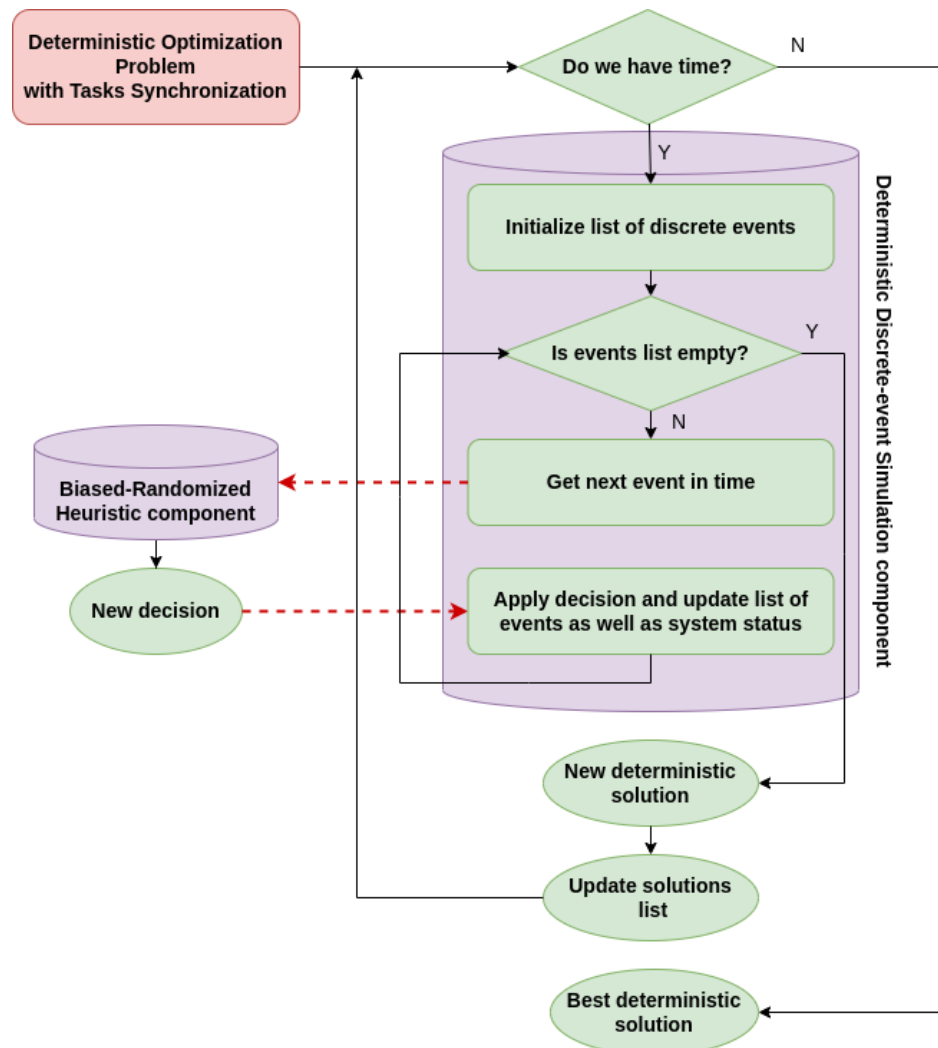


Figure 4. Flow-chart illustrating the DEH concept.

#### 4. Illustrative Example

In this section, an illustrative example is described. We consider the application of a DEH algorithm to a shuttle–lift–crane based automated storage and retrieval system (SLC-AS/RS) [27]. The SLC-AS/RS is an automated warehouse frequently used in the steel industry, where it is generally dedicated to billets or metal bar bundles storage. The storage and retrieval operations involve three different types of machines: shuttles, lifts, and cranes. Each machine is autonomous, but it has to cooperate with other machines. Because of this requirement, the system is characterized by a high degree of synchronization in the tasks of the machines. An accurate description of an SLC-AS/RS is provided in Zammori et al. [28], Bertolini et al. [29] and Neroni [27], who provide a good description of the system elements (Figure 5).

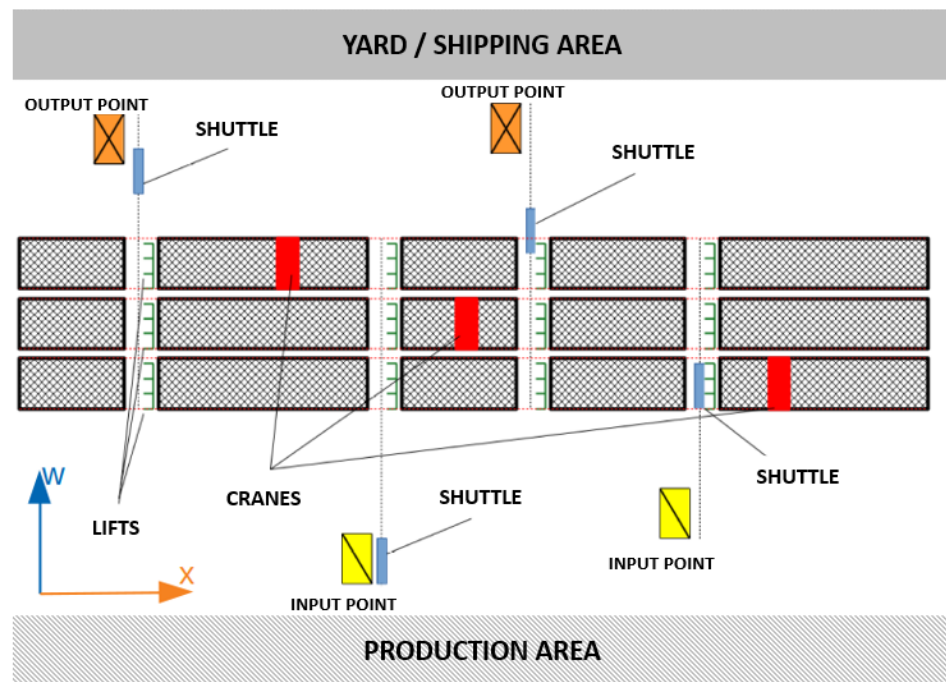


Figure 5. Schematic representation of SLC-AS/RS elements.

The optimization of the SLC-AS/RS productivity involves several decisions and many different aspects. On the one hand, the entering material must be assigned to a storage location that can host it, and the assignment to a specific storage location causes certain machines to be involved. On the other hand, customers’ orders are characterized by three elements: the product required, the quantity required—expressed in kilograms—and the minimum quality level required. According to these three characteristics, the system has to define which items in stock are retrieved to fulfill each customer order. Retrieving one item instead of another might have a significant impact on the operations schedule.

The optimization of a system like this requires two different components: (i) a discrete event simulation process to correctly reproduce the system behavior, including all synchronizations and machines interoperability; and (ii) a heuristic optimization component able to take all the above-mentioned decisions by providing a high-quality solution in a short computational time. In this way, the DES component can be used to evaluate the effects of the decisions made by the heuristic procedure, and the heuristic component can use the DES response to make its next decisions. In order to provide an illustrative example, the heuristic component is simply a greedy approach that emulates the ‘common sense’ decisions an expert manager would make. The greedy approach is then enhanced with a biased randomization behavior in order to generate a different solution at each iteration, thus allowing exploring the solution space to escape from local optima. A simplified pseudo-code of the proposed solution is illustrated in Algorithm 1.

---

**Algorithm 1** The proposed solution approach based on the DEH

---

```

bestSol ← heuristic()
bestMakespan ← simulation(bestSol)
while availableComputationalTime > 0 do
    newSol ← heuristic()
    newMakespan ← simulation(newSol)
    if newMakespan < bestMakespan then
        bestSol ← newSol
        bestMakespan ← newMakespan
    end if
end while

```

---

It is important to point out that using the DES provides the framework with many further benefits in terms of flexibility and simplicity. First, it would be possible to consider aspects that are typical in a real-world scenario (e.g., machine downtimes and failures). Secondly, the DES could be extended into a simheuristic approach through the introduction of stochastic operations with only slight changes in the overall framework. Finally, the simulation may provide additional information to define if a solution is better than another, such as the overall machine waiting time, the length of queues, etc. In Figure 6, the results of the BR-DEH algorithm are compared to the ones provided by a simple greedy heuristic. The results are expressed in terms of the makespan needed to carry out 20 storage and retrieval requests (hence, lower is better). The SLC-AS/RS used for in the example is the one presented in Figure 5. It consists of two shuttles exclusively dedicated to input operations, two shuttles dedicated to output operations, three cranes (one on each storage rack), and 12 lifts. Results in Figure 6 show the efficiency of the BR-DEH, and the computational times displayed in Figure 7 prove that this approach can be implemented in a real-life scenario.

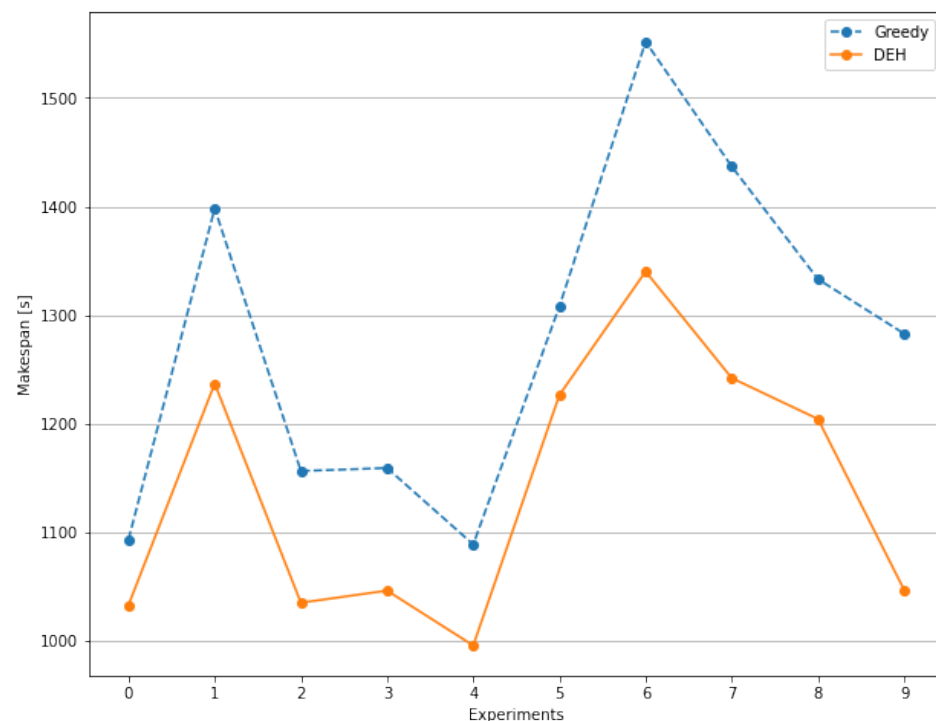


Figure 6. Results of the DEH on the SLC-AS/RS.

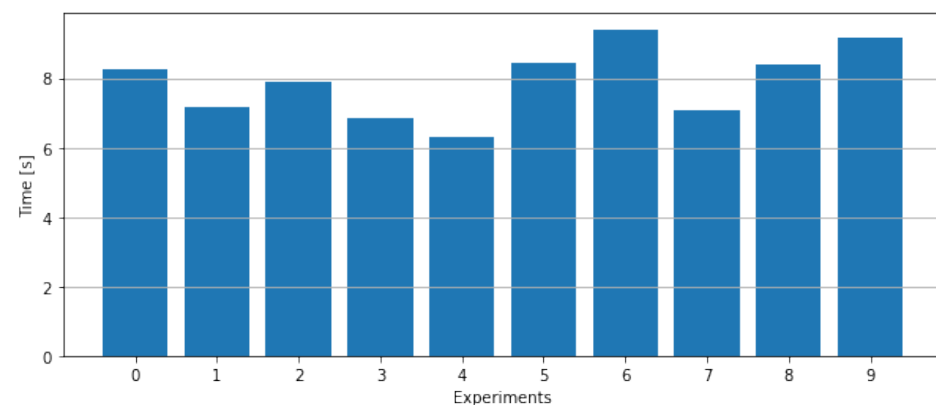


Figure 7. Computational times of the DEH on the SLC-AS/RS.



## 5. Optimization Problems with Synchronization Issues

Problems with synchronization have attracted much attention in research related to the behavior of complex networks. The ability of a network to synchronize depends on many factors and parameters associated with its structure [30]. There are many examples in the industry today where event synchronization plays an important role in ensuring the efficient functioning of a network or system. Generally, the application of BR-DEH algorithms targets all those problems characterized by synchronization issues. These scenarios are typically characterized by three aspects: (i) a high coefficient of dynamism; (ii) a limited number of elements or resources; and (iii) a set of events whose triggering and succeeding depend on each other. In particular, the triggering of an event can depend on the succeeding of one or more other events, and the succeeding of an event may give rise to other events not scheduled before. In these cases, the integration of DES in a biased-randomized algorithm is essential to model the system and verify with consistency and accuracy the effects of each decision made by the heuristic [31]. In this section, we give an overview of the main optimization problems with synchronization issues, where the implementation of the BR-DEH strategy is suggested as a natural and effective solving procedure.

In manufacturing, flow-shop and job-shop optimization problems might require the implementation of a BR-DEH in scenarios characterized by dependencies on machines, resources, and unpredictable inconveniences such as machine failures and down-times. For instance, it is frequent to have a flow shop in which the processing time on a machine depends on the jobs that the machine processed before. This dependency is usually because of tool wear and setup times. In job shops, there are situations in which machines require operators (or resources). A limited number of operators (or resources) are shared between all machines, moving from one machine to another according to their necessity. Both flow shops and job shops can be solved using a BR-DEH when influenced by unpredictable events, such as machine failures or the arrival of unscheduled jobs. An example of a real problem encountered in the literature is processing different jobs by a series of machines modeled as a hybrid flow-shop problem with several additional and realistic constraints [32]. Another example is on-demand production systems when a workflow has parallel processes, specific machine loops or re-entry cycles, in which jobs may re-enter specific processes at some point in the flow-shop chain [33]. Similarly, semiconductor manufacturing can be modeled as a hybrid flow-shop problem with time dependencies and priority constraints [34].

In logistics and transportation, vehicle routing problems with synchronization are addressed and associated with different real-life cases. The adoption of BR-DEH algorithms is suggested when the interference among vehicles is not negligible. This is true for transportation problems related to last-mile delivery networks [35–37], omnichannel vehicle routing [31], or home healthcare services [38,39] with traffic and precedence constraints, among others [40]. They are also seen in internal logistics when optimizing paths traveled by AGVs [41]. This involves many well-known operations research problems, such as path finding, minimum spanning tree, traveling salesman problem, and many else. Transportation problems usually face situations characterized by sudden events, such as changes in delivery points or requests and necessities to reorganize the routes. The high dynamism makes them a suitable target for BR-DEH algorithms. A realistic example studied by Arnau et al. [42] shows the dynamism of modern transport systems. They present the problem of transporting containers through interconnected networks. Containers are transported from origin to their final destinations on or before a given time frame and may be temporarily stored at network hubs. Each truck carries one container at a time, containers may be transported by different trucks during their journey from origin to destination, and drivers have to be back at their starting points in due time.

Telecommunication networks are another field in which optimization challenges involve synchronization issues in their structure. They require strict frequency and time synchronization between neighboring base stations to accelerate data transfer without overflow, underflow, bit errors, or other adverse effects. This occurs in code division multiple access, global system

for mobile communications, and universal mobile telecommunications systems. In global positioning systems, the use of signals is also conditioned by the location of antennas to obtain a clear view of the sky [43]. For these reasons, operators are constantly searching for a better synchronization scheme that allows for sharing very accurate timing information reliably and cost-effectively [44]. Al-Makhadmeh and Tolba [45] study one of the most popular problems in telecommunications, the unit-response (or master–slave) configuration, in which there is a leader (master) and several devices (slaves) that need to synchronize with the leader. This problem is known as master–slave synchronization of chaotic systems. In secure communications, for instance, there is an unknown input—the transmitted information—that is only retrieved for transmission if and only if the master and slave devices are correctly synchronized. Similar situations characterized by the three aspects mentioned above (i.e., high dynamism, limited shared resources, and dependent events) can also be found in cloud computing [46]. Even in this case, limited resources (e.g., servers or endpoints), unexpected requests and the fact that events depend on each other make this problem a perfect target for BR-DEH algorithms.

Other fields in which dynamic optimization problems with synchronization issues are encountered, and where BR-DEH algorithms might be helpful, are those involving different agents whose actions depend on each other. For example, problems considering automated storage and retrieval (AS/RS) systems (Section 4). In these systems, a single operation involves many different machines capable of exchanging the unit loads. These machines can be seen as resources in the case of the job shop, since they are limited, shared by different operations, and dependent on each other. Even in this case, when two or more machines share the same path, we need to model interferences among machines (as in the case of AGVs). A similar situation concerns ride-sharing and car-sharing scenarios [47], where other traffic constraints and unexpected requests might be considered to provide solutions suitable for being implemented in real life. In addition, in genetics, optimization problems with synchronization issues occur. This is the case of the so-called consensus genetic mapping of species, which is based on different sources of mapping information. The challenge arises from inter-population differences in the rate of recombination and the distribution of exchanges along chromosomes, variations in the dominance of the markers used, and the use of different subsets of markers in different laboratories. Thus, for instance, Ronin et al. [48] model it as a traveling salesman problem with synchronization. For each data set, the goal was to search for the order that gives the minimum sum of recombination distances between adjacent markers.

Therefore, given the current availability of data on different real-life networks (e.g., biological, informational, social, etc.) and with complex infrastructures, there can be many variants of optimization problems with synchronization issues associated with the occurrence of different events. This is because many networks have features such as noise-generating disruptions, evolving or changing connections, and fixed structures, but the activation of nodes depends on a related factor. In general, regardless of the field of application, BR-DEH algorithms can be used to solve dynamic optimization problems with synchronization issues, which typically involve one or more agents that depend on the actions or results that other agents generate.

## 6. Existing Work on DEH Algorithms

After describing the DES approach and using it to model and simulate complex systems, Fikar et al. [10] noticed that it could be combined with heuristics to solve NP-Hard optimization problems with synchronization requirements. This section summarizes some contributions utilizing BR-DEH algorithms. Thus, the aforementioned authors proposed a solution approach for a home service routing problem. In the home service routing problem, operators are assigned several customer homes to visit, and vehicles distribute them among homes. Because of new placed service requests and cancellations, home assignments and routes are dynamic and should be updated during the day. The solution approach supports planners in assigning operators and defining vehicle routes dynamically.

In addition, this approach benefits from a defined event list to construct vehicle routes and operator-homes assignments during the day, which allows for minimizing the number of required vehicles. Similarly, Bayliss et al. [31] used a BR-DEH approach to solve the omnichannel vehicle routing problem. In the problem, a retail store is replenished, and online-customers orders should be picked up and delivered. The solution approach utilizes the BR-DEH methodology to construct solutions that are refined using a local search operator. The approach found promising solutions for most of the problem instances. Another transportation problem is solved by Arnau et al. [42]. In their problem, a container should be delivered on or before a given deadline. This container is transported along a network of hubs by trucks. One truck could transport only one container, and the containers could be stored temporarily at any hub. A BR-DEH algorithm is employed to solve the problem by specifying which containers are selected to be temporarily transported from a hub, as well as which containers have to be stored at hubs until future notice. Then, a biased-randomized behavior is introduced into the heuristic to construct several variants of the deterministic solution. In order to propose an approach to handle the problem, an iterated local search framework integrates the biased-randomized heuristic to find the most promising solutions in short computation times.

As discussed in Section 5, problems in manufacturing are good candidates to be solved using the BR-DEH approach. For example, Laroque et al. [32] use this methodology to solve a flow-shop problem in the semiconductor industry. Jobs are processed by machines based on job type, and these jobs could be batched along the production line. In an extension to the problem, job priorities of being processed on machines in the flow-shop problem are considered. Similarly, Laroque et al. [34] used a DEH algorithm to recommend a solution that minimizes the production makespan. In addition, the solution approach utilizes biased randomization to construct several solutions and Monte Carlo simulation to evaluate the solutions. Another manufacturing problem involves the re-entry and re-processing of parts on machines. This problem considers re-entry cycles of jobs due to quality issues. Thus, the same machine that processed the job in the first cycle should handle it for the second cycle. This problem is modeled as a flow-shop problem, and Juan et al. [33] defined a discrete-event list to manage the defined constraints in the problem. In addition, the authors integrated a biased-randomized procedure to find a solution in a short computational time.

A dynamic ride-sharing problem arises in the context of edge computing in smart cities and the integration of the Internet of Things. The dynamic ride-sharing problem refers to one of the routing problems related to smart cities, in which a driver uses a private car to pick up riders along a route. However, after departing from the origin (after defining the initial route), new ride requests and traffic conditions might influence decisions regarding defined routes. Thus, the routes should be dynamically updated with the changes along the route. This problem is solved using a multi-start approach integrating BR-DEH to construct solutions [49].

## 7. Cross-Problem Analysis of Computational Results

This section presents a summary of results previously published in different works available in the literature. Specifically, this section shows the computational results obtained using BR-DEH algorithms to solve several dynamic optimization problems with time dependencies. On the one hand, Table 1 reports the chosen papers and the references used to gather the computational results. On the other hand, Table 2 presents the solution values reported by the different authors for the listed problems. The first column identifies the reference from where the solution values have been gathered, while the second column identifies the problem. Afterwards, the following two columns report the Best Known Solution for the specific problem (BKS) and the solution obtained using the BR-DEH approach (OBS), respectively. Finally, the last column shows the percentage gap between OBS and BKS.

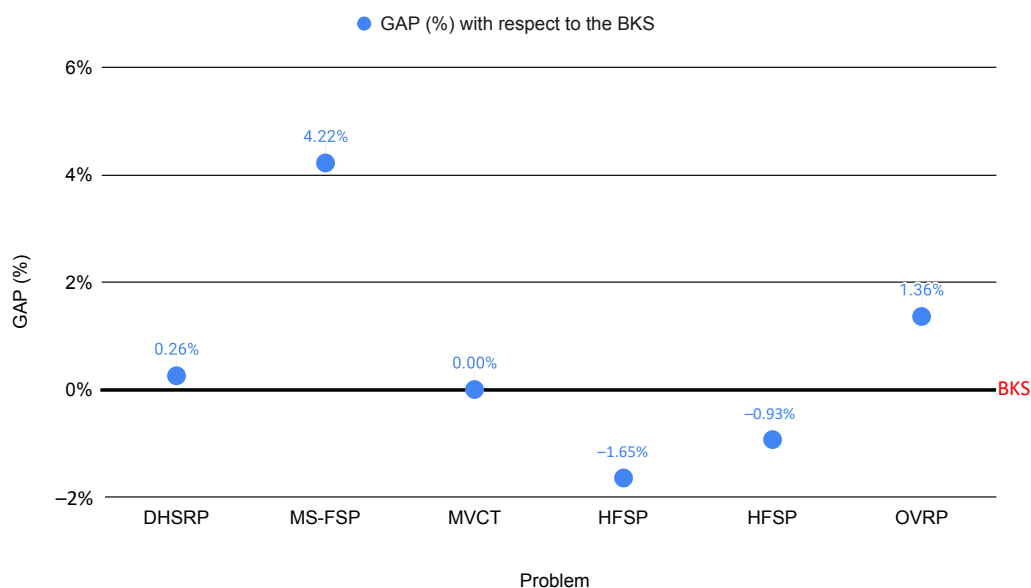
**Table 1.** Selected optimization problems.

Problem	Acronym	Reference
Dynamic Home Service Routing Problem	DHSRP	[10]
Multi-Server Flow-Shop Problems with Machine Re-Entering	MS-FSP	[33]
Multi-Vehicle Container Transport	MVCT	[42]
Hybrid Flow-Shop Problem with Time Dependencies	HFSP	[34]
Hybrid Flow-Shop Problem with Batching and Multiple Paths	HFSP	[32]
Omnichannel Vehicle Routing Problem	OVRP	[31]

**Table 2.** Information on the selected problems.

Reference	Problem	BKS	OBS	GAP (%)
Fikar et al. [10]	DHSRP	895.0	897.3	0.26%
Juan et al. [33]	MS-FSP	1336.3	1392.7	4.22%
Arнау et al. [42]	MVCT	8.4	8.4	0.00%
Laroque et al. [34]	HFSP	10,079.8	9913.6	−1.65%
Laroque et al. [32]	HFSP	13,975.1	13,844.6	−0.93%
Bayliss et al. [31]	OVRP	520.2	527.2	1.36%

The gaps between the different solutions for the analyzed works are presented in Figure 8, where the *y*-axis represents the gap between OBS and BKS. According to the results, BR-DEH algorithms provide good performance, improving on average in about −0.62% the BKS and varying from about 4.22% for the MS-FSP up to −1.65% for the HFSP. Concerning the MS-FSP, which presents the highest gap, authors compare the obtained results using the classical single-server flow-shop problem without loops. Moreover, they argue that BKS was obtained using a complete metaheuristic that uses analytical expressions to compute the solution, together with local search operators and data structures that allow for more efficient solution space exploration. Unfortunately, these metaheuristics cannot be employed in general scenarios with multi-server machines and loops due to their time dependencies. Thus, a BR-DEH algorithm is necessary to solve these problems efficiently.



**Figure 8.** Gaps between OBS and BKS (baseline 0% gap).

### 8. Conclusions and Future Work

This paper has reviewed the concept of biased-randomized discrete-event heuristics, which hybridizes biased-randomization techniques with discrete-event simulation concepts

with the purpose of solving dynamic optimization problems with time dependencies or synchronization issues. The manuscript also explains why the BR-NEH approach can effectively solve complex dynamic optimization problems with time dependencies, which can be frequently found in real-life applications. The way in which the different components interact in a BR-DEH has also been discussed, emphasizing the different stages that can contribute to make the approach more efficiently while searching for high-quality solutions that satisfy the synchronization issues.

Recent applications of BR-DEH algorithms in fields such as logistics and transportation or manufacturing and production have also been analyzed, and a numerical summary of previous works illustrating the capabilities of BR-DEH methods to provide high-quality solutions to different dynamic problems is also provided.

There are several lines of research that are still open in the field of BR-DEH algorithms. Among these, we can highlight the following ones: (i) the introduction of machine learning methods that allow us to predict the evolution of the system, and, therefore, help us to make better decisions in the long term; (ii) the efficient and easy integration of BR-DEH code developed with modern programming languages with commercial simulators like FlexSim, which currently supports a friendly interaction with Python; and (iii) the extension of BR-DEH algorithms into simheuristics [50,51], so they can also deal with stochastic versions of the dynamic optimization problems.

**Author Contributions:** Conceptualization, A.A.J. and M.A.; methodology, J.C. and M.N.; software, J.P.; writing—original draft preparation, J.C., M.N., M.A. and J.P.; writing—review and editing, A.A.J. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work has been partially funded by the Spanish Ministry of Science (PID2019-111100RB-C21-C22 / AEI/10.13039/501100011033), the Barcelona City Council and Fundació “la Caixa” under the framework of the Barcelona Science Plan 2020-2023 (grant 21S09355-001) and the Generalitat Valenciana (PROMETEO/2021/065).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Kuehn, W. Digital twins for decision-making in complex production and logistic enterprises. *Int. J. Des. Nat. Ecodynamics* **2018**, *13*, 260–271. [[CrossRef](#)]
2. Roodbergen, K.J.; Vis, I.F. A survey of literature on automated storage and retrieval systems. *Eur. J. Oper. Res.* **2009**, *194*, 343–362. [[CrossRef](#)]
3. Gagliardi, J.P.; Renaud, J.; Ruiz, A. Models for automated storage and retrieval systems: A literature review. *Int. J. Prod. Res.* **2012**, *50*, 7110–7125. [[CrossRef](#)]
4. Schneier, M.; Schneier, M.; Bostelman, R. *Literature Review of Mobile Robots for Manufacturing*; National Institute of Standards and Technology, US Department of Commerce: Gaithersburg, MD, USA, 2015.
5. Mehami, J.; Nawari, M.; Zhong, R.Y. Smart automated guided vehicles for manufacturing in the context of Industry 4.0. *Procedia Manuf.* **2018**, *26*, 1077–1086. [[CrossRef](#)]
6. Illgen, S.; Höck, M. Literature review of the vehicle relocation problem in one-way car sharing networks. *Transp. Res. Part Methodol.* **2019**, *120*, 193–204. [[CrossRef](#)]
7. Eren, E.; Uz, V.E. A review on bike-sharing: The factors affecting bike-sharing demand. *Sustain. Cities Soc.* **2020**, *54*, 101882. [[CrossRef](#)]
8. Mitropoulos, L.; Kortsari, A.; Ayfantopoulou, G. A systematic literature review of ride-sharing platforms, user factors and barriers. *Eur. Transp. Res. Rev.* **2021**, *13*, 1–22. [[CrossRef](#)]
9. Lee, S.; Smart, M.J.; Golub, A. Difference in travel behavior between immigrants in the us and us born residents: The immigrant effect for car-sharing, ride-sharing, and bike-sharing services. *Transp. Res. Interdiscip. Perspect.* **2021**, *9*, 100296. [[CrossRef](#)]
10. Fikar, C.; Juan, A.A.; Martinez, E.; Hirsch, P. A Discrete-event Driven Metaheuristic for Dynamic Home Service Routing with Synchronised Trip Sharing. *Eur. J. Ind. Eng.* **2016**, *10*, 323–340. [[CrossRef](#)]
11. Goldsman, D.; Goldsman, P. Discrete-event simulation. In *Modeling and Simulation in the Systems Engineering Life Cycle*; Springer: London, UK, 2015; pp. 103–109.

12. Desale, S.; Rasool, A.; Andhale, S.; Rane, P. Heuristic and meta-heuristic algorithms and their relevance to the real world: A survey. *Int. J. Comput. Eng. Res. Trends* **2015**, *351*, 2349–7084.
13. Juan, A.A.; Corlu, C.G.; Tordecilla, R.D.; de la Torre, R.; Ferrer, A. On the use of biased-randomized algorithms for solving non-smooth optimization problems. *Algorithms* **2019**, *13*, 8. [[CrossRef](#)]
14. Rubinstein, R.Y.; Melamed, B. *Modern Simulation and Modeling*; Wiley New York: New York, NY, USA, 1998; Volume 7.
15. Kumar, R.; Garg, V.K. *Modeling and Control of Logical Discrete Event Systems*; Springer Science & Business Media: New York, NY, USA, 2012; Volume 300.
16. Rasmussen, M.S.; Justesen, T.; Dohn, A.; Larsen, J. The home care crew scheduling problem: Preference-based visit clustering and temporal dependencies. *Eur. J. Oper. Res.* **2012**, *219*, 598–610. [[CrossRef](#)]
17. Law, A.M. *Simulation Modeling and Analysis*, 5th ed.; Mcgraw-Hill: New York, NY, USA, 2014.
18. Banks, J.; Carson, J.S., II; Nelson, B.L.; Nicol, D.M. *Discrete-Event System Simulation*, 5th ed.; Pearson: London, UK, 2005.
19. Robinson, S. *Simulation: The Practice of Model Development and Use*; Bloomsbury Publishing: London, UK, 2014.
20. Zhang, X. Application of discrete event simulation in health care: A systematic review. *BMC Health Serv. Res.* **2018**, *18*, 687. [[CrossRef](#)] [[PubMed](#)]
21. Hamrock, E.; Paige, K.; Parks, J.; Scheulen, J.; Levin, S. Discrete event simulation for healthcare organizations: A tool for decision-making. *J. Healthc. Manag.* **2013**, *58*, 110–124. [[CrossRef](#)]
22. Ahalt, V.; Argon, N.T.; Ziya, S.; Strickler, J.; Mehrotra, A. Comparison of emergency department crowding scores: A discrete-event simulation approach. *Health Care Manag. Sci.* **2018**, *21*, 144–155. [[CrossRef](#)] [[PubMed](#)]
23. Demirli, K.; Al Kaf, A.; Simsekler, M.C.E.; Jayaraman, R.; Khan, M.J.; Tuzcu, E.M. Using lean techniques and discrete-event simulation for performance improvement in an outpatient clinic. *Int. J. Lean Six Sigma* **2021**, *12*, 1260–1288. [[CrossRef](#)]
24. Nogueira, G.P.M.; de Assis Rangel, J.J.; Croce, P.R.; Peixoto, T.A. The environmental impact of fast delivery B2C e-commerce in outbound logistics operations: A simulation approach. *Clean. Logist. Supply Chain.* **2022**, *5*, 100070. [[CrossRef](#)]
25. de Bittencourt, G.C.; Chagas, R.D.S.; Silva, V.A.; Vianna, I.G.P.; Longhi, R.P.; Ribas, P.C.; Ferreira Filho, V.J.M. A solution framework for the integrated problem of cargo assignment, fleet sizing, and delivery planning in offshore logistics. *Comput. Ind. Eng.* **2021**, *161*, 107653. [[CrossRef](#)]
26. Alvanchi, A.; Baniassadi, F.; Shahsavari, M.; Kashani, H. Improving materials logistics plan in road construction projects using discrete event simulation. *Eng. Constr. Archit. Manag.* **2021**, *28*, 3144–3163. [[CrossRef](#)]
27. Neroni, M. Improvement of Logistics Automation: A Focus on Unconventional Solutions. Ph.D. Thesis, Università degli Studi di Parma, Dipartimento di Ingegneria e Architettura, Parma, Italy, 2021.
28. Zammori, F.; Neroni, M.; Mezzogori, D. Cycle time calculation of shuttle-lift-crane automated storage and retrieval system. *IISE Trans.* **2021**, *54*, 40–59. [[CrossRef](#)]
29. Bertolini, M.; Esposito, G.; Mezzogori, D.; Neroni, M. Optimizing Retrieving Performance of an Automated Warehouse for Unconventional Stock Keeping Units. *Procedia Manuf.* **2019**, *39*, 1681–1690. [[CrossRef](#)]
30. Nishikawa, T.; Motter, A.E. Maximum performance at minimum cost in network synchronization. *Phys. D Nonlinear Phenom.* **2006**, *224*, 77–89. [[CrossRef](#)]
31. Bayliss, C.; Martins, L.d.C.; Juan, A.A. A two-phase local search with a discrete-event heuristic for the omnichannel vehicle routing problem. *Comput. Ind. Eng.* **2020**, *148*, 106695. [[CrossRef](#)]
32. Laroque, C.; Leißau, M.; Copado, P.; Panadero, J.; Juan, A.A.; Schumacher, C. A biased-randomized discrete-event heuristic for the hybrid flow shop problem with batching and multiple paths. In Proceedings of the 2021 Winter Simulation Conference (WSC), Phoenix, AZ, USA, 12–15 December 2021; pp. 1–11.
33. Juan, A.A.; Copado, P.; Panadero, J.; Laroque, C.; de la Torre, R. A discrete-event heuristic for makespan optimization in multi-server flow-shop problems with machine re-entering. In Proceedings of the 2020 Winter Simulation Conference (WSC), Orlando, FL, USA, 14–18 December 2020; pp. 1492–1502.
34. Laroque, C.; Leißau, M.; Copado, P.; Schumacher, C.; Panadero, J.; Juan, A.A. A Biased-Randomized Discrete-Event Algorithm for the Hybrid Flow Shop Problem with Time Dependencies and Priority Constraints. *Algorithms* **2022**, *15*, 54. [[CrossRef](#)]
35. Sarasola, B.; Doerner, K.F. Adaptive large neighborhood search for the vehicle routing problem with synchronization constraints at the delivery location. *Networks* **2020**, *75*, 64–85. [[CrossRef](#)]
36. Kitjacharoenchai, P.; Min, B.C.; Lee, S. Two echelon vehicle routing problem with drones in last mile delivery. *Int. J. Prod. Econ.* **2020**, *225*, 107598. [[CrossRef](#)]
37. Moshref-Javadi, M.; Hemmati, A.; Winkenbach, M. A truck and drones model for last-mile delivery: A mathematical model and heuristic approach. *Appl. Math. Model.* **2020**, *80*, 290–318. [[CrossRef](#)]
38. Haddadene, S.R.A.; Labadie, N.; Prodhon, C. A GRASP × ILS for the vehicle routing problem with time windows, synchronization and precedence constraints. *Expert Syst. Appl.* **2016**, *66*, 274–294. [[CrossRef](#)]
39. En-nahli, L.; Afifi, S.; Allaoui, H.; Nouaouri, I. Local search analysis for a vehicle routing problem with synchronization and time windows constraints in home health care services. *IFAC-PapersOnLine* **2016**, *49*, 1210–1215. [[CrossRef](#)]
40. Tomar, I.; Sreedevi, I.; Pandey, N. State-of-Art Review of Traffic Light Synchronization for Intelligent Vehicles: Current Status, Challenges, and Emerging Trends. *Electronics* **2022**, *11*, 465. [[CrossRef](#)]
41. Jiang, M.; Huang, G.Q. Intralogistics synchronization in robotic forward-reserve warehouses for e-commerce last-mile delivery. *Transp. Res. Part E Logist. Transp. Rev.* **2022**, *158*, 102619. [[CrossRef](#)]

42. Arnau, Q.; Barrena, E.; Panadero, J.; de la Torre, R.; Juan, A.A. A biased-randomized discrete-event heuristic for coordinated multi-vehicle container transport across interconnected networks. *Eur. J. Oper. Res.* **2022**, *302*, 348–362. [[CrossRef](#)]
43. Han, J.; Jeong, D.K. Practical considerations in the design and implementation of time synchronization systems using IEEE 1588. *IEEE Commun. Mag.* **2009**, *47*, 164–170.
44. Ahmed, F.; Asghar, M.Z.; Imran, A. Combinatorial Optimization for Artificial Intelligence Enabled Mobile Network Automation. In *Metaheuristics in Machine Learning: Theory and Applications*; Springer: Cham, Switzerland, 2021; pp. 663–690.
45. Al-Makhadmeh, Z.; Tolba, A. An intelligence-based recurrent learning scheme for optimal channel allocation and selection in device-to-device communications. *Circuits Syst. Signal Process.* **2020**, *39*, 997–1018. [[CrossRef](#)]
46. Pawlikowski, K.; Jeong, H.D.; Lee, J.S. On credibility of simulation studies of telecommunication networks. *IEEE Commun. Mag.* **2002**, *40*, 132–139. [[CrossRef](#)]
47. Stiglic, M.; Agatz, N.; Savelsbergh, M.; Gradisar, M. Enhancing urban mobility: Integrating ride-sharing and public transit. *Comput. Oper. Res.* **2018**, *90*, 12–21. [[CrossRef](#)]
48. Ronin, Y.; Mester, D.; Minkov, D.; Belotserkovski, R.; Jackson, B.; Schnable, P.; Aluru, S.; Korol, A. Two-phase analysis in consensus genetic mapping. *G3 Genes Genomes Genet.* **2012**, *2*, 537–549. [[CrossRef](#)]
49. Peyman, M.; Copado, P.J.; Tordecilla, R.D.; Martins, L.D.C.; Xhafa, F.; Juan, A.A. Edge Computing and IoT Analytics for Agile Optimization in Intelligent Transportation Systems. *Energies* **2021**, *14*, 6309. [[CrossRef](#)]
50. Gök, Y.S.; Padrón, S.; Tomasella, M.; Guimarans, D.; Ozturk, C. Constraint-based robust planning and scheduling of airport apron operations through simheuristics. *Ann. Oper. Res.* **2022**, 1–36. [[CrossRef](#)]
51. Yazdani, M.; Mojtahedi, M.; Loosemore, M. Enhancing evacuation response to extreme weather disasters using public transportation systems: A novel simheuristic approach. *J. Comput. Des. Eng.* **2020**, *7*, 195–210. [[CrossRef](#)]