



**Please cite the Published Version**

Guia, Sana Sahar, Laouid, Abdelkader, Hammoudeh, Mohammad , Bounceur, Ahcène, Alfawair, Mai and Eleyan, Amna  (2022) Co-simulation of multiple vehicle routing problem models. Future Internet, 14 (5). 137 ISSN 1999-5903

**DOI:** <https://doi.org/10.3390/fi14050137>

**Publisher:** MDPI AG

**Version:** Published Version

**Downloaded from:** <https://e-space.mmu.ac.uk/634411/>

**Usage rights:**  [Creative Commons: Attribution 4.0](https://creativecommons.org/licenses/by/4.0/)

**Additional Information:** This is an open access article which first appeared in Future Internet, published by MDPI

**Enquiries:**

If you have questions about this document, contact [openresearch@mmu.ac.uk](mailto:openresearch@mmu.ac.uk). Please include the URL of the record in e-space. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from <https://www.mmu.ac.uk/library/using-the-library/policies-and-guidelines>)



## Article

# Co-Simulation of Multiple Vehicle Routing Problem Models

Sana Sahar Guia <sup>1</sup>, Abdelkader Laouid <sup>1</sup>, Mohammad Hammoudeh <sup>2</sup>, AHCène Bounceur <sup>3,\*</sup>,  
Mai Alfawair <sup>4</sup> and Amna Eleyan <sup>5</sup>

<sup>1</sup> LIAP Laboratory, University of El Oued, El Oued 39000, Algeria; sanasahar-guia@univ-eloued.dz (S.S.G.); abdelkader-laouid@univ-eloued.dz (A.L.)

<sup>2</sup> Information & Computer Science Department, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia; m.hammoudeh@kfupm.edu.sa

<sup>3</sup> Lab-STICC UMR CNRS, University of Western Brittany UBO, 6285 Brest, France

<sup>4</sup> Faculty of Information Technology, AlBalqa Applied University, Amman 11134, Jordan; mai.alfauri@bau.edu.jo

<sup>5</sup> School of Computing and Mathematics, Manchester Metropolitan University, Manchester M1 5GD, UK; a.eleyan@mmu.ac.uk

\* Correspondence: ahcene.bounceur@univ-brest.fr; Tel.: +33-651122408

**Abstract:** Complex systems are often designed in a decentralized and open way so that they can operate on heterogeneous entities that communicate with each other. Numerous studies consider the process of components simulation in a complex system as a proven approach to realistically predict the behavior of a complex system or to effectively manage its complexity. The simulation of different complex system components can be coupled via co-simulation to reproduce the behavior emerging from their interaction. On the other hand, multi-agent simulations have been largely implemented in complex system modeling and simulation. Each multi-agent simulator's role is to solve one of the VRP objectives. These simulators interact within a co-simulation platform called MECASYCO, to ensure the integration of the various proposed VRP models. This paper presents the Vehicle Routing Problem (VRP) simulation results in several aspects, where the main goal is to satisfy several client demands. The experiments show the performance of the proposed VRP multi-model and carry out its improvement in terms of computational complexity.

**Keywords:** complex system; multi-simulation; vehicle routing problem



**Citation:** Guia, S.S.; Laouid, A.; Hammoudeh, M.; Bounceur, A.; Alfawair, M.; Eleyan, A.

Co-Simulation of Multiple Vehicle Routing Problem Models. *Future Internet* **2022**, *14*, 137. <https://doi.org/10.3390/fi14050137>

Academic Editors: Agostino Poggi, Martin Kenyeres, Ivana Budinská and Ladislav Hluchy

Received: 30 March 2022

Accepted: 27 April 2022

Published: 29 April 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

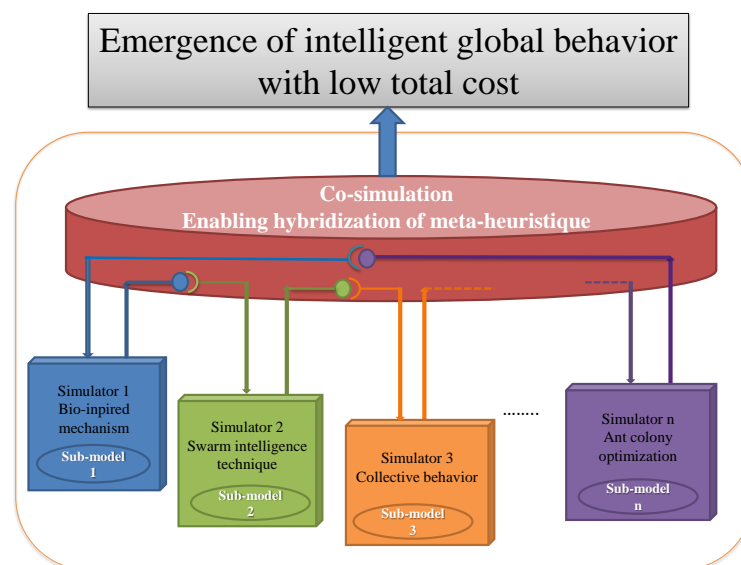
Complex systems is a multidisciplinary field and is the subject of active research in several domains such as physics, biology, social sciences and cognitive sciences. The authors of [1–3] define a complex system as a set of heterogeneous entities that interact between them, where each entity is characterized by its cooperative, adaptive and open nature. Simulation is one of the critical tools frequently used to study complex systems [4]. Simulation may reduce experimentation costs by allowing researchers to test various alternatives and scenarios of complex systems [5].

The Vehicle Routing Problem (VRP) presents a complex system [6,7] such that the VRP is considered as a group of dispersed clients that are served by a fleet of vehicles that start and end at one depot [8]. Recently, VRP and its various models attracted the interest of intelligent transport researchers due to the complementary limitations of real-world problems, the potential cost savings and the possibility of service improvement in distributed systems. Due to the intrinsic complexity of determining rigorous models and optimal solutions to a large-scale instance of VRP, the process that involves exact solvers is limited, difficult and time-consuming [9]. Therefore researchers and transportation companies are interested in meta-heuristics such as Ant Colony Optimization (ACO), Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) to solve real world VRP by producing close to optimal solution. An efficient simulation may offer an opportunity

to improve the performance of VRP by modeling different methods for solving various VRP objectives.

Co-simulation is applied in many different multidomain systems [10]. It is proposed to address the issues and challenges of complex systems [11]. In this context, decomposed components of a complex system are modeled by different sub-models, then simulated using a simulator for each sub-model. Co-simulation enables exchanging coupling connections to achieve the behavior of the whole complex system as shown in Figure 1. The co-simulation middleware named Multi-agent Environment for Complex System CO-simulation (MECSYCO) allows the integration of several modeling and simulation software to co-simulate the dynamic behavior of a complex system. MECSYCO makes the possibility to compare simulation results, swap, add, or remove models.

This paper aims to address the challenge of showing the impact of simulating a VRP multi-model as several interacting simulators by reducing the traveled distance and the obtained time during vehicles routing with the aim to satisfy clients demands. In this context, we will design and implement multiple VRP models as complex systems with several perspectives of multi agent simulators. Each simulator in the proposed solution solves different objectives of the VRP and then makes them interact into the co-simulation platform. Multi agent simulators that represent the VRP models are developed using Netlogo and the whole system is reproduced based on the MECSYCO co-simulator. We observed that the interacting VRP models provide high quality solutions by minimizing significantly the cost of services offered to the clients, in terms of computational time and travel distance.



**Figure 1.** Co-simulation of coupled system.

The rest of this work is organized as follows: In Section 2, we present the summary of the related multi-simulation of complex system research and introduce an overview of the used platform in the co-simulation and we describe the problem formulation of the VRP. The proposed solution is presented in Section 3. Section 4 provides the implementation details and the experiments results. Section 5 concludes the work and give some future work avenues.

## 2. Related Work

To better simulate the behavior of the whole complex and dynamic system, multi-modeling and co-simulation are used to provide a strategy that collects a set of models from different discipline [12]. The modular architecture in the multi-agent environment of complex system co-simulation middleware offers the integration of heterogeneous

simulators and models. In the literature, several works such as [5,13], have been proposed for the simulation of the complex system as interacting models for reproducing the behavior of the entire system. The work of [14] captures the improvement of decomposing and integrating continuous systems using MECSYCO as discrete environment. In [15], the authors designed the co-simulation of complex system with integral multi-agent simulators, while MECSYCO was used in [16] to integrate the modeling and the simulation tools to co-simulate the complete cyber-physical system.

The hybridization of meta-heuristics in the multi-agent system is used to facilitate the development of meta-heuristic frameworks' optimization. In addition, the development of hybrid meta-heuristics is flexible in multi-agent systems and it offers simultaneous exploration of various areas of the research space [17].

Several studies have been performed to solve different variations of VRP by using multi-agent systems [18]. The work of [19] uses the reinforcement learning and pattern matching to provide a multi-agent distributed framework where each agent adapts itself. In [20], the authors solve the VRP with Time Windows (VRPTW) by exploiting the principle of reinforcement learning to improve the agent actions based on the solutions generated by other agents and the environment interactions.

To improve the result of VRP, this paper aims to create an interacting VRP model by combining the multi-agent simulators to reproduce the entire VRP multi-model. From the literature, we observe that there is no proposal in multi agent systems that integrates simulators into a co-simulation platform to reproduce the entire VRP multi-model.

### 2.1. Co-Simulation of a Complex System

MECSYCO has been used as a middle-ware to simulate and model complex systems which allows the utilization of existing simulators for the implementation of heterogeneous and numerical simulations. MECSYCO focuses on both, the DEVS common formalism proposed by [21] that uses a discrete-event abstraction for the design of dynamic system [22], and on the Agent and artifact architecture using a multi-agent concepts [13] to perform the heterogeneous co-simulation of complex system. In this case, each model or simulator has been considered as an agent and the interaction between simulators corresponds to the indirect cooperation between agents using the defined concept of artifact in [23]. Coupling artifact aims to determine the input output connections between two models and to create the model agent (m-agent) which uses the artifact for leading their model and interchanging input/output data. MECSYCO manipulates a model as various composed models in interactions without a global coordinator using a simulator for each model [24]. This step occupies the management of decentralized multi model simulation in order to integrate several aspects of the same system into a coherent one and to deal with the main challenges of modeling and simulation of a complex system. On the other hand, MECSYCO already includes integrated simulators such as multi agent platform Netlogo, which is frequently used for simulating complex systems and modeling natural and social phenomena.

### 2.2. Mathematical Formulation of the Problem

This section deals with the mathematical formulation of the problem. In the combinatorial optimization field, VRP is one of the most challenging research problems. The issue was studied more than 40 years ago. It involves defining the best sequence of routes for a fleet of vehicles to provide the service to a specific set of clients. The fleet of vehicles is situated at a central depot, every vehicle has certain capacity and each client has a given demand. The aim is to optimize the total traveled distance to serve a geographically dispersed group of clients [25]. First, we have the following data:

1. One depot;
2. A set of commands;
3. A fleet of vehicles.

The objective is to reduce the total traveled distance from a depot to serve clients, using a similar fleet of vehicles. The issue of solving the VRP faces several design challenges and

focuses on the assignment of clients to the routes. This process involves the determination of several routes for each depot by (1) assigning each client to a unique route without exceeding the capacity of the vehicle, and (2) determining a series of clients on every vehicle route. Hence, the mathematical formulation of the vehicle routing problem can be described as follows:

- $R$  denotes the set of nodes,  $R = \{r_0, r_1, r_2, \dots, r_n\}$ , with  $r_0$  is the depot and  $r_i, i \neq 0$  is the client;
- $V$  denotes the sets of vehicles;
- $N$  is the number of all clients;
- $M$  is the number of vehicles;
- $Q(k)$  is the capacity of vehicle  $k$ ;
- $E$  is the edge set between nodes defined as:  $E = \{(r_i, r_j) | i \neq j \wedge r_i, r_j \in R\}$ ;
- $b_i$  is the demand of client  $i$ ;
- $c_{ij}$  is the cost of transporting one unit from node  $i$  to node  $j$ ;
- $x_{ij}^k$  is 1, if vehicle  $k$  travels from node  $i$  to node  $j$ ; 0, otherwise,
- $y_{ik}$  is 1, if vehicle  $k$  offers service for client  $i$ ; 0, otherwise.

From these definitions the objective function problem can be formulated as follows:

$$\min \sum_{i \in R} \sum_{j \in R} c_{ij} \sum_{k \in V} x_{ij}^k \quad (1)$$

where Equation (1) represents the objective function that aims to reduce the total travel cost and the constraints formulations are:

$$\sum_{j \in R} x_{0j}^k = 1, \forall k \in V \quad (2)$$

$$\sum_{j \in R, j \neq i} \sum_{k \in V} x_{ij}^k = 1, \forall i \in R \quad (3)$$

$$\sum_{i \in R} b_i y_{ik} \leq Q(k), \forall k \in V \quad (4)$$

$$\sum_{i \in R} x_{il}^k = \sum_{j \in R} x_{lj}^k, \forall l \in R, \forall k \in V \quad (5)$$

where Equation (2) aims to ensure that every client is allocated in the single route and Equation (3) is the capacity constraint set for vehicles. Whereas Equation (4) is defined to assure that each client can be served only once and Equation (5) ensures that all delivery vehicles must return to the original depot after finishing their task.

### 3. The Designed Model

The goal of this paper is to solve the VRP by building a complex simulation using a society of interacting co-evolving VRP models. The proposed approach provides a solution based on the agent and artifact concept [23,26,27]. The simulation is performed by m-agents that manage their model and data exchange using artifacts.

The first task to do is to define a solution for solving each VRP model using an atomic VRP simulator. Then, the second task is to determine the input output connections between these simulators. The final task is to construct coupled interacting co-evolving VRP models.

#### 3.1. Description of Atomic VRP Models

The routing and scheduling of a service correspond to the creation of vehicle routes for the depot. By assigning each client to a unique route and respecting the capacity of the vehicle the order of the clients on every vehicle route will be determined. To this end, we develop two models to solve VRP such that each model is to satisfy a different objective.

The first is to try to minimize the total traveling distance and the second is to minimize the number of vehicles.

### 3.1.1. First VRP Model

We use ant colony optimization for the VRP, where the colony of ants is created from the depot. Each Ant constructs a route for vehicles that will serve the clients starting from the depot and returning to the same depot. The principle of the routing algorithm is described below, see Algorithm 1.

---

#### Algorithm 1 Routing and sequencing of the first model

---

**Require:**  $R, ant\_colony, Q(k)$ ;  
**Ensure:**  $route\_map$ ;  
1: **for** ( $a \in ant\_colony$ ) **do**  
2:    $route\_map \leftarrow r_0$ ;  
3:    $R \leftarrow R - r_0$ ;  
4:    $k \leftarrow 0$ ;  
5:    $Q(k) \leftarrow 0$ ;  
6:    $i \leftarrow r_0$ ;  
7:   **while**  $R \neq \emptyset$  **do**  
8:     **for** ( $r_j \in R$ ) **do**  
9:        $p_{ij} \leftarrow probability\_of\_point\_i\_to\_choose\_point\_j$ ;  
10:     **end for**  
11:      $r_s \leftarrow r_j\_with\_highest\_p_{ij}$ ;  
12:      $Q(k) \leftarrow Q(k) + demand(r_s)$ ;  
13:     **if**  $Q(k) > capacity\_of\_vehicle$  **then**  
14:        $k \leftarrow k + 1$ ;  
15:        $Q(k) \leftarrow 0$ ;  
16:        $route\_map \leftarrow route\_map + r_0$ ;  
17:        $i \leftarrow r_0$ ;  
18:     **else**  
19:        $route\_map \leftarrow route\_map + r_s$   
20:        $i \leftarrow r_s$   
21:        $R \leftarrow R - r_s$ ;  
22:     **end if**  
23:   **end while**  
24: **end for**  
25: **return**  $route\_map\_of\_the\_best\_ant\_solution$ ;

---

The improvements of the solution shown in Algorithm 1 reached in successive iterations. In each iteration all ants in the colony travel between a pair of clients. The probability of an ant to visit an unvisited client is calculated based on the pheromone and the distance and a client will be chosen according to this probability. Then the ant checks if it can visit the given client, i.e., if its current load allows it to add the demand of the client and verify the vehicle's capacity violation. If yes, the ant visits the specified client (adding client's demand and indicating it as selected). Otherwise, the ant ends the current trip with empty load (returning to its colony) and the algorithm restarts a new trip. We apply the following equations for a client  $i$  to choose the next client  $j$

$$p_{ij} = \frac{(\tau_{ij})^\alpha \times (\eta_{ij})^\beta}{\sum_{l \in R} (\tau_{il})^\alpha \times (\eta_{il})^\beta} \quad (6)$$

$$\eta_{ij} = \frac{1}{cost_{ij}} \quad (7)$$

where  $p_{ij}$  is the probability of the point  $i$  to choose point  $j$ ,  $\tau_{ij}$  is the strength of pheromone trail between point  $i$  and  $j$ ,  $\alpha$  is the coefficient that controls the influence of the pheromone

trail  $\tau_{ij}$ ,  $\beta$  is the coefficient that controls the influence of the visibility  $\eta_{ij}$ , and  $R$  is a set of clients.

At the end of each iteration, the pheromone trail evaporates according to the following equation:

$$\tau_{ij} = (1 - \rho) \times \tau_{ij} \quad (8)$$

where  $\rho$  is the coefficient of the pheromone evaporation and the update state is calculated by applying the next equation:

$$\tau_{ij} = \tau_{ij} + \Delta\tau_{ij} \quad (9)$$

where

$$\Delta\tau_{ij} = \frac{\text{best\_known\_solution}}{\text{best\_global\_cost}} \quad (10)$$

### 3.1.2. Second VRP Model

VRP of the second model uses a similar algorithm routing and sequencing of the first model. The difference is that each ant in this model tries to reduce the number of vehicles to visit all clients and to add all possible ones whose demand does not exceed the capacity of the current vehicle. The algorithm of the second model is formulated as follows, see Algorithm 2.

---

#### Algorithm 2 Routing and sequencing of the second model

---

**Require:**  $R, ant\_colony, Q(k)$ ;  
**Ensure:**  $route\_map$ ;  
1: **for** ( $a \in ant\_colony$ ) **do**  
2:    $route\_map \leftarrow r_0$ ;  
3:    $R \leftarrow R - r_0$ ;  
4:    $C \leftarrow R - r_0$ ;  
5:    $k \leftarrow 0$ ;  
6:    $Q(k) \leftarrow 0$ ;  
7:    $i \leftarrow r_0$ ;  
8:   **while**  $R \neq \emptyset$  **do**  
9:     **while**  $C \neq \emptyset$  **do**  
10:       **for** ( $r_j \in C$ ) **do**  
11:           $p_{ij} \leftarrow \text{probability\_of\_point}_i\text{\_to\_choose\_point}_j$ ;  
12:       **end for**  
13:        $r_s \leftarrow r_j\text{\_with\_highest\_}p_{ij}$ ;  
14:        $C \leftarrow C - r_s$ ;  
15:        $Q(k) \leftarrow Q(k) + \text{demand}(r_s)$ ;  
16:       **if**  $Q(k) < \text{capacity\_of\_vehicle}$  **then**  
17:           $route\_map \leftarrow route\_map + r_s$   
18:           $i \leftarrow r_s$   
19:           $R \leftarrow R - r_s$ ;  
20:       **end if**  
21:     **end while**  
22:      $k \leftarrow k + 1$ ;  
23:      $Q(k) \leftarrow 0$ ;  
24:      $route\_map \leftarrow route\_map + r_0$ ;  
25:      $i \leftarrow r_0$ ;  
26:   **end while**  
27: **end for**  
28: **return**  $route\_map\_of\_the\_best\_ant\_solution$ ;

---

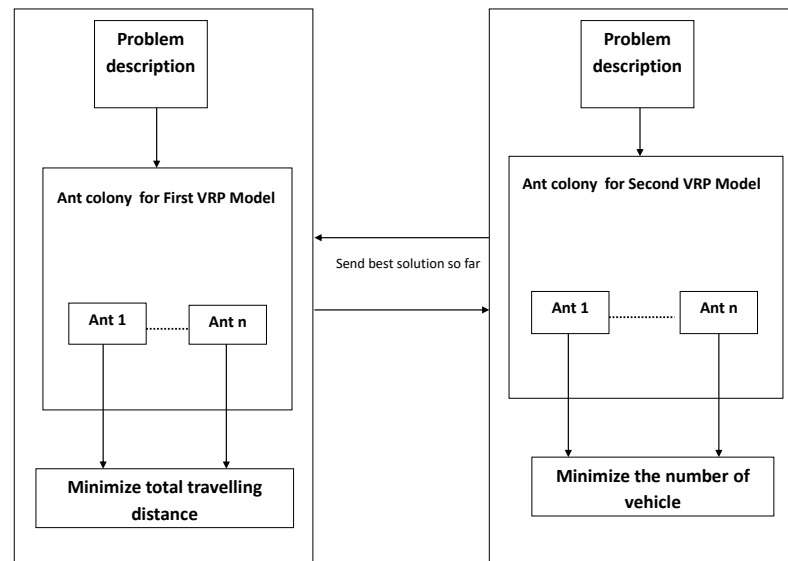
In Algorithm 2, the ant checks if its current load allows it to add the demand of the chosen client and verifies the vehicle's capacity violation. If yes, the ant visits the specified client (adding client's demand and indicating it as selected). Otherwise, the ant selects another unvisited node according to the calculated probability. When all clients have been



visited, the ant ends the current trip with empty load (returning to its colony) and the algorithm begins a new trip. We improve the solution by local search and we use a two-opt local search algorithm in the end of both Algorithms 1 and 2.

### 3.2. Structuring VRP Multi-Model

In order to co-simulate the multi-model of VRP, we use MECSYCO, which allows to lunch of several VRP models. In this approach, we focus on the reuse of atomic VRP models defined in Section 3.1 and making them interact. Figure 2 shows an overview on the model coupling of  $M_1$  and  $M_2$ .



**Figure 2.** Model dependency overview.

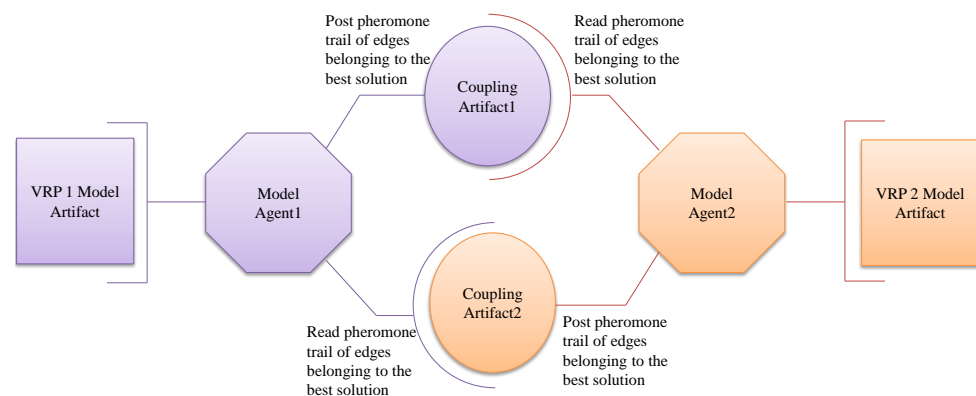
As VRP models are created separately, each model has its own description and proceeds its own execution. It should couple these VRP models to reconstruct the entire multi-model using the co-simulation of MECSYCO. To this end, we define the connections between both models by specifying the input and output ports.

#### 3.2.1. Exchanging Data between Simulators: Input and Output Connections

To predict the impact of co-simulation of VRP multi-model, we show the first model  $M_1$  which represents the solution of VRP using Algorithm 1, where the objective is to minimize the total traveled distance. The second model  $M_2$  implements Algorithm 2 that tries to minimize the total number of vehicles used to solve the same VRP. The aim in this case is to reproduce the VRP multi-model as a society of interacting and co-evolving models. We have to couple the two aforesaid models  $M_1$  and  $M_2$  using the concepts of Agent and Artifacts that are presented in Section 1. To couple  $M_1$  and  $M_2$  we need to define the input and output ports of each model and the connections between them.

The solution is improved by successive iterations where every model, i.e.,  $M_1$  or  $M_2$ , shares the improved solution with the other.  $M_1$  and  $M_2$  read the input and try to improve their solution as illustrated in Figure 3.





**Figure 3.** Coupling VRP models.

Specifically, when the model  $M_1$  finds new best solution in a given iteration, it sends the following inputs to the model  $M_2$

1.  $Ed$ : Set of edges belonging to the best solution (for each one of these edges the model  $M_1$  sends a pair of nodes delimiting this edge).
2.  $P$ : Set of pheromones trail of each edge belonging to the best solution and the model  $M_2$  retrieves these values of pheromone and modify its own values to have the new ones sent by  $M_1$ .

These sent pheromones trail to the model  $M_2$  will modify the pheromone trail of the same edges in  $M_2$  for improving the best solution finding. According to the ant colony system, in Algorithms 1 and 2 the ant chooses the next node to be added to the solution according to

- The cost of the best solution represented by the total traveled distance;
- The edge with minimum distance to be traveled;
- The edge with high value of pheromone trail.

The edge with high value of pheromone trail have a high probability to be added to the solution. Here, the model  $M_2$  have edges with elevated values of pheromone trail from:

- Those of its own best solution;
- Those of  $M_1$ 's best solution.

In the same way, when model  $M_2$  finds a new best solution, it sends its best solution to model  $M_1$  and replaces the current value of pheromone trail in this model by those of the new best solution in model  $M_2$ . Figure 4 show more details on the behavior of models in interaction.

### 3.2.2. Interchanging Models

When executing two similar models of VRP, they give two different solutions in each iteration. Instead of coupling two different VRP models, we can simply integrate two similar models  $M_1$  or two similar models  $M_2$ . Here, we keep the same input and output connections as defined in the previous section. So, if one model has a good solution, it sends it to the second model and reciprocally. Specifically, when we couple two same models  $M_1$ , they will have different solutions in each iteration, and the first one have better solution, send the following inputs to the second model  $M_1$ :

1.  $Ed$ : Set of edges belonging to the best solution.
2.  $P$ : Set of pheromones trail of each edge belonging to the best solution and the second model  $M_1$  retrieves these values of pheromone and modify its own values to have the new ones sent by the first model  $M_1$ .

This may impact on the solution of each model and redirect it to the best solution by reinforcing the pheromone of edges that have a good solution in each model. Moreover, we can study the evolution of the computational time when proceeding the co-simulation of VRP models, thereby the co-simulation of VRP models for finding a good solution.

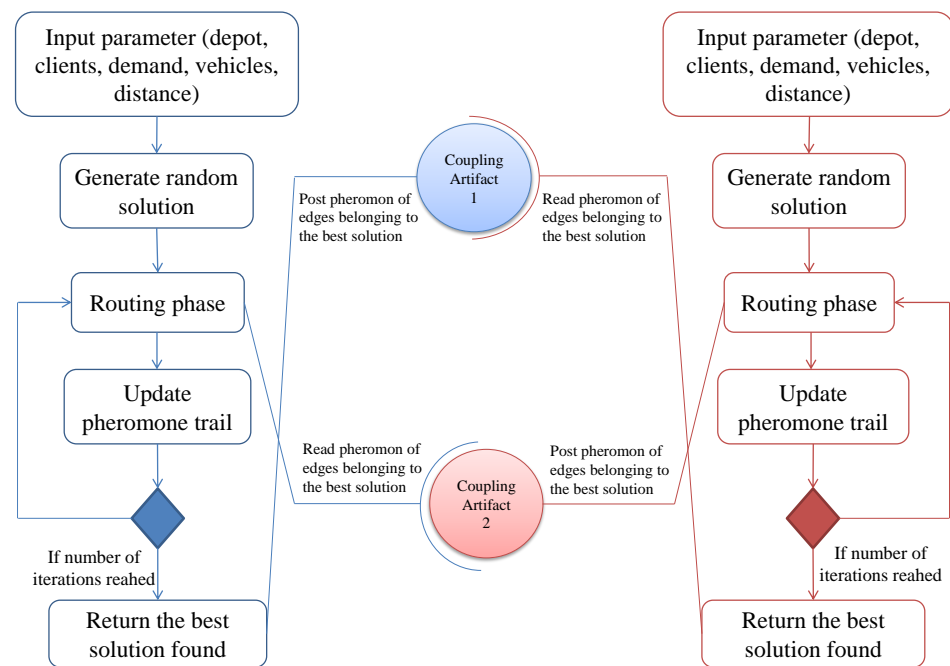


Figure 4. Behavior of two models in interaction.

Thus, we interchange models to compare different executions and study the accuracy of the best multi model. Figure 5 shows the VRP models' interaction graph.

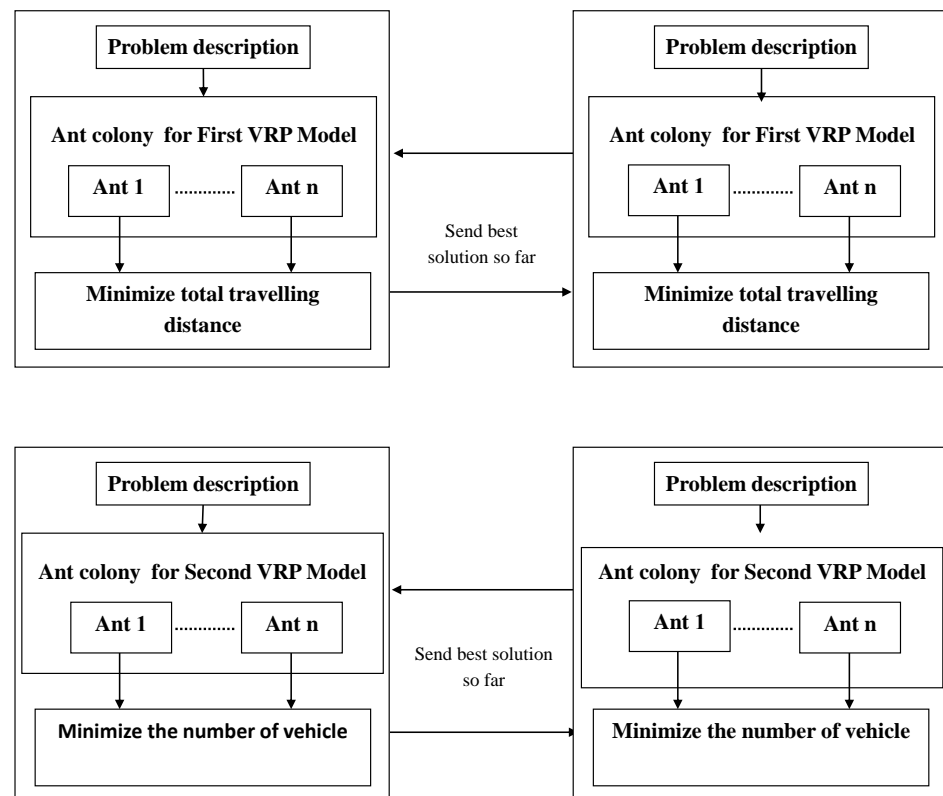


Figure 5. Graph of VRP interaction models.

By using MECSYCO we may launch more than one model of VRP at the same time following the routing algorithm defined in Section 2. The execution gives two different

solutions that are improved in each iteration. The whole architecture of the multi-model is modular, transparent and parallel so the simulation can be viewed as a set of distributed and reusable components such that:

- The modeling is performed by model artifact.
- Conducting the models is accomplished by the model-agents.
- The coordination process is carried out by the coupling-artifacts.

As a result, we easily can add, remove or interchange models without being interested with coupling and coordination issues.

#### 4. Implementation and Results

The VRP models are implemented using Netlogo [28] which is an environment designed for modeling and simulating the natural and social experiences. We use MECASYCO to perform the co-simulation of the VRP models based on the following parameters of hardware:

- Memory: 7.7 GB;
- CPU: Intel® Core™ i5-8250U CPU @ 1.60GHz × 8;
- Graphic Card: Intel® UHD Graphics 620 (KabyLake GT2);
- Operating System: ubuntu 16.04 LTS 64 bits;
- Disk: 964 GB.

In the analysis, we study the behavior and the experiment results of VRP models on several VRP instances from [29] and then we evaluate the improvement carried out by the co-simulation of the VRP multi-model in term of quality of obtained results and computational time.

##### 4.1. Implementing VRP Models

In the definition and implementation of the VRP graphs using Netlogo, we use the turtles agent to represent nodes with their turtle variables. The coordinate of each client and the links agent are used to define the edge between two nodes. We use the link variables for determining for each edge the link cost of these two nodes and the value of pheromone trail.

Table 1 represents the result obtained by executing single model VRP: model  $M_1$  with BKS is the best-known solutions which were given by [29], Perc-Dev is the percentage deviation of the result reached by the model  $M_1$  and Processing time (s) is the processing time for the execution of the model  $M_1$ .

**Table 1.** VRP results of  $M_1$ .

Inst.	BKS	Model 1	Per-Dev %	Processing Time (s)
A-n32-k5	784	794	1.25	16.261
A-n33-k5	661	661	0	26.46
A-n33-k6	742	759	2.23	11.98
A-n60-k9	1408	1442	2.35	91.18
B-n78-k10	1266	1332	4.95	112.18

Table 2 represents the result obtained by executing single model VRP, i.e., model  $M_2$ . Both models  $M_1$  and  $M_2$  are based on an ant colony system in their implementation, and they can interchange the following data: the cost of solution is calculated by the total traveled distance, edges belonging in the best solution found and the values of pheromone trail of these edges.

Other data can be exchanged between these models such as the route of the best solution or the route of each vehicle, etc. However, we skip these exchanged data to avoid more machine resources occupation because we must reserve more memory space to store the solution (routes) which takes a considerable time during the exchange.

**Table 2.** VRP result of  $M_2$ .

Inst.	BKS	Model 2	Per-Dev %	Processing Time (s)
A-n32-k5	784	807	2.85	19.241
A-n33-k5	661	661	0	38.84
A-n33-k6	742	757	1.98	16.82
A-n60-k9	1408	1491	5.56	90.83
B-n78-k10	1266	1318	3.94	1043.46

#### 4.2. Co-Simulation of VRP Multi-Model

In this section, we run different forms of combined models and perceive the improvement provided by their co-simulation. In order to develop interactive VRP multi-models, we use MECSYCO which offers multiple integrating VRP models by defining the input and output port of coupling artefact. VRP instances [29] are a set of 100 instances created to offer a more universal and stable experimental setting.

Experiments were carried out in order to analyze the performance of the proposed VRP multi-model. The main objective is to assess whether the co-simulation has a direct impact on the qualitative performance of the obtained results. In this context, three composition are proposed for the interaction between the VRP multi-model used in the experiments:

- Coupling two different VRP models:  $M_1$  with  $M_2$ .
- Coupling two same VRP models:  $M_1$  with  $M_1$ .
- Coupling two same VRP models:  $M_2$  with  $M_2$ .

Tables 3–5 show the result of VRP co-simulation where:

- Symbol A means: after co-simulation;
- Symbol B means: before co-simulation;
- Perc-Dev B is the percentage deviation of single model before co-simulation;
- Perc-Dev A is the percentage deviation after co-simulation for two models in interaction;
- Processing time B (s) is the processing time in second for just one model before co-simulation;
- Processing time A (ms) is the processing time in milliseconds after co-simulation for two models in interaction.

Table 3 presents the co-simulation results of model  $M_1$  with  $M_1$  which demonstrate enhanced performance of the found solution in terms of percentage deviation and processing time compared with the execution of single model  $M_1$ .

**Table 3.** The obtained co-simulation result of models  $M_1$  with  $M_1$ .

Inst.	Per-Dev B %	Per-Dev A %	Processing Time B (s)	Processing Time A (ms)
A-n32-k5	1.25	0.63	16.26	381
A-n33-k5	0	0	26.46	377
A-n33-k6	2.23	1.32	11.98	160
A-n60-k9	2.35	1.53	91.18	162
B-n78-k10	4.95	2.98	112.18	739

Table 4 shows the result of the combination of the two same models  $M_2$  with  $M_2$ . We perceive that there is an improvement in the found solution compared with one model  $M_2$  in both the solution cost and the time of processing.

**Table 4.** The obtained co-simulation result of models  $M_2$  with  $M_2$ .

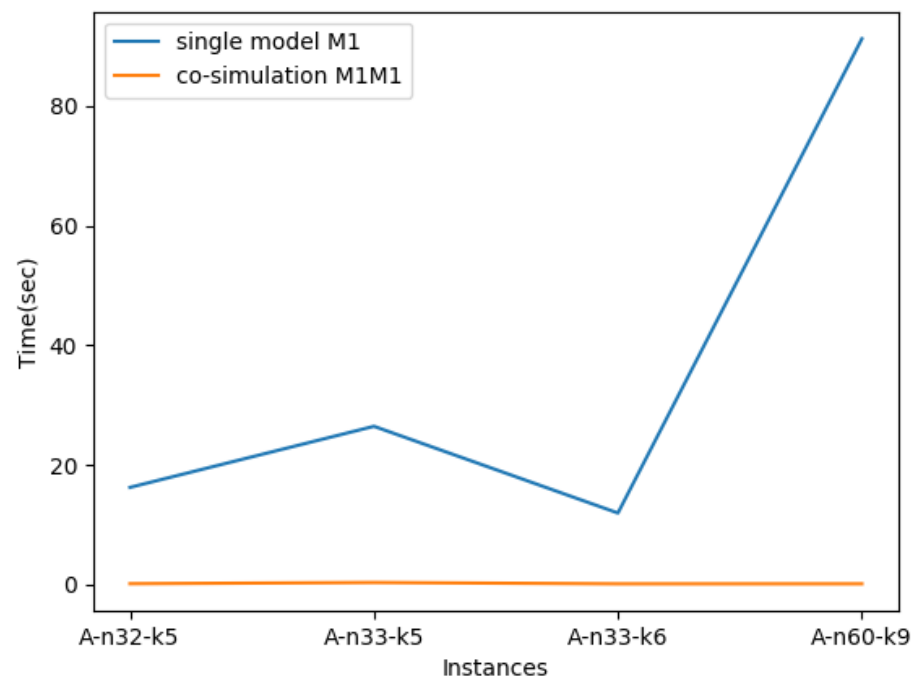
Inst.	Per-Dev B %	Per-Dev A %	Processing Time B (s)	Processing Time A (ms)
A-n32-k5	2.85	0.63	19.24	959
A-n33-k5	0	0	38.84	774
A-n33-k6	1.98	2.11	16.82	310
A-n60-k9	5.56	2.89	90.83	2242
B-n78-k10	3.94	0.62	1043.46	2633

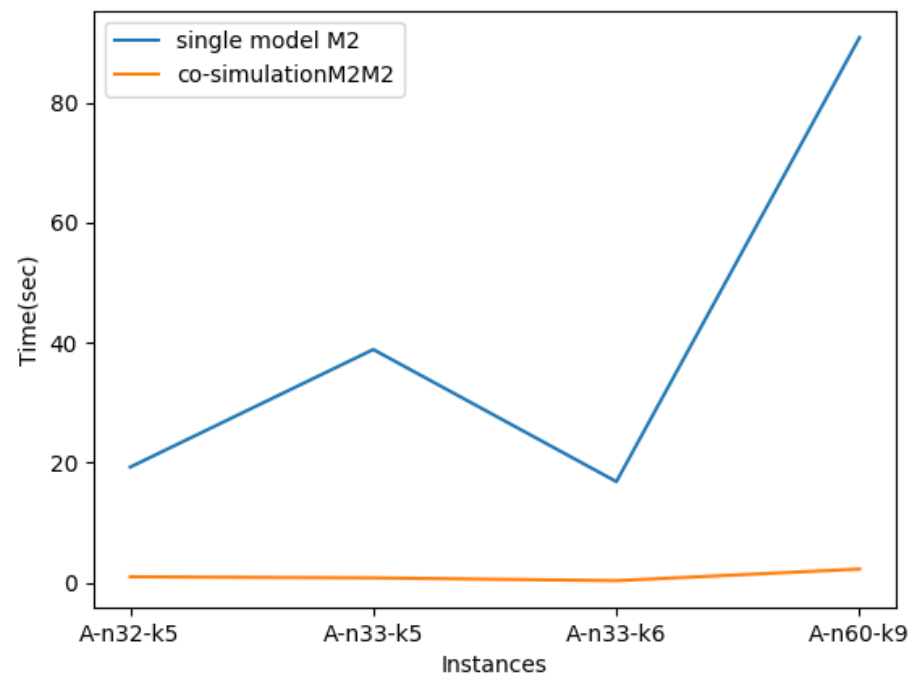
Table 5 shows the obtained results of integrating two different models  $M_1$  with  $M_2$ . We observe that there is an improvement in the found solution compared with the result of a single model  $M_1$  or  $M_2$  in terms of solution and time of processing.

**Table 5.** The obtained co-simulation result of models  $M_1$  with  $M_2$ .

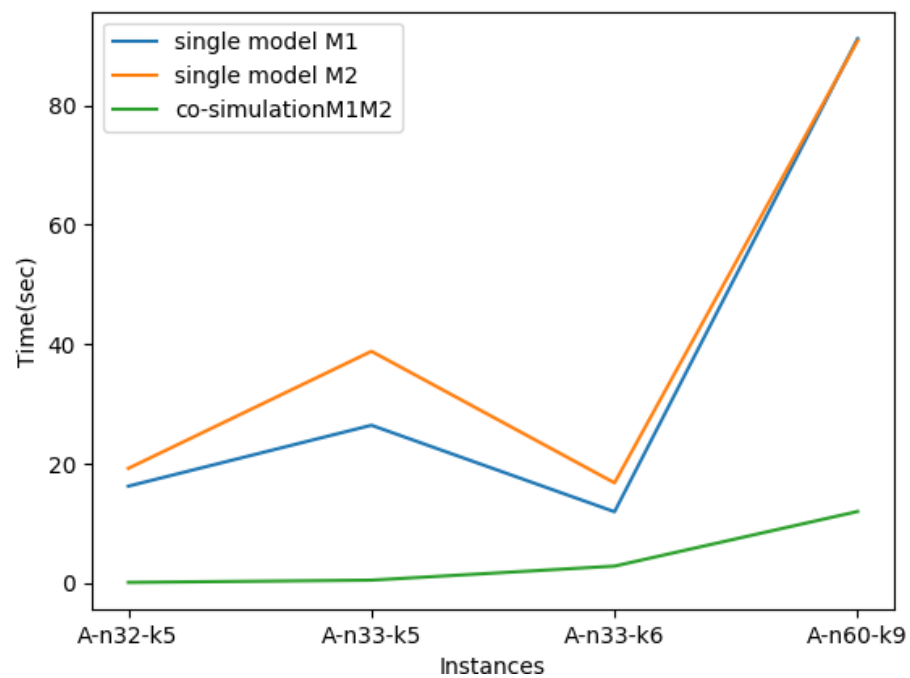
Inst.	Per-Dev B %	Per-Dev A %	Processing Time B (s)	Processing Time A (ms)
A-n32-k5	2.85	1.63	16.26	152
A-n33-k5	0	0	26.46	525
A-n33-k6	2.23	1.46	11.98	413
A-n60-k9	2.35	1.88	91.18	2877
B-n78-k10	4.95	3.28	112.18	12,022

Figures 6 and 7 show the co-simulation processing time when coupling the same or different models. We conclude that the co-simulation of models is faster than the unique model  $M_1$  or  $M_2$ . On the other hand, during data exchange between models we proceed to post and read best solutions for each interacting model. Instead of sending the best solution as a sequence of routes, we choose to send it as separate edges with their pheromone trail. After the execution, we observe that this action has a significant impact on the behavior of the model in constructing its solution and the model builds better solutions quickly. In addition, it does not spend a lot of time interchanging data and occupies a reasonable memory size to store the best solution.

**Figure 6.** Cont.



**Figure 6.** The execution time of co-simulation models compared with single model  $M_1$  or  $M_2$ .



**Figure 7.** The execution time of co-simulation models  $M_1$  with  $M_2$ .

#### 4.3. Discussion

We conducted several experiments in a single VRP model and multiple VRP models in interaction. Single VRP models implementing different algorithms of ACO metaheuristics gave close to optimal solutions; the VRP multi-model improved these results compared with single ones. We coupled

- Two different models  $M_1$  and  $M_2$ ;
- Two same models  $M_1$  with  $M_1$ ;
- Two same  $M_2$  with  $M_2$ .

Always the result of coupled models was the best according to single model  $M_1$  or single model  $M_2$ . It was especially time consuming.

Figure 8 displays the co-simulation processing time of the three coupled systems and shows that the coupled system using two same models  $M_1$  has the best result.

Co-simulation enables the integration of heterogeneous models/simulators homogeneously, in the future we can implement other types of metaheuristic and define the appropriate interchanging data between them, in order to study the amelioration of results in each different case.

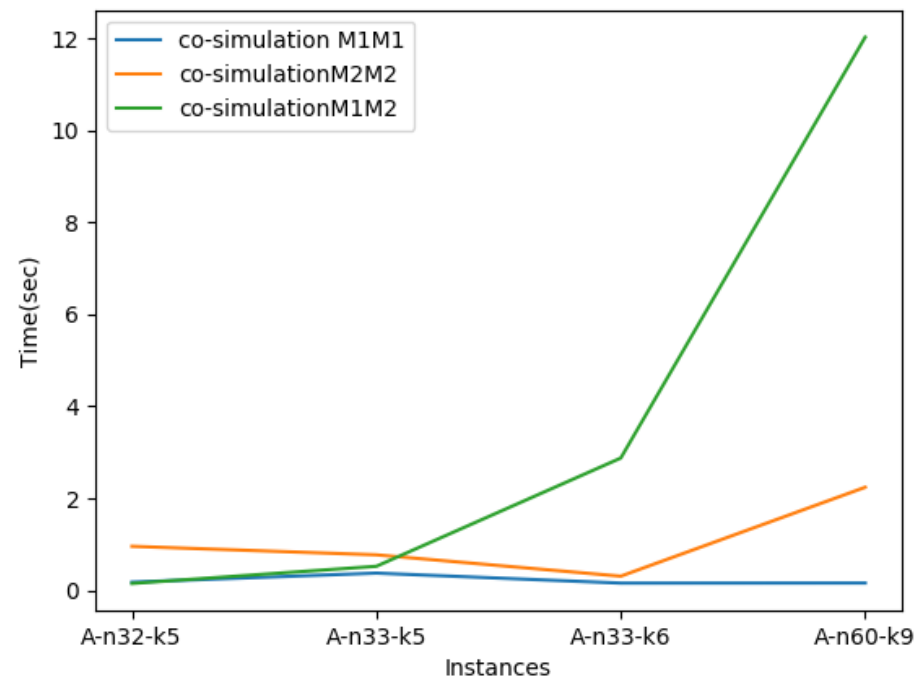


Figure 8. The execution time of co-simulation models .

## 5. Conclusions

This paper discusses the integration of VRP simulators created in Netlogo platform into MECASYCO co-simulation middleware. The proposed VRP multi-models in interaction implement ant colony meta-heuristic for solving two different objectives. The obtained results show the efficiency of better solution finding of VRP multi-models compared with single VRP models in terms of processing time and travel distance. We conclude that the type of data exchanged between models gives more qualitative performance to the solution short execution time to solve the problem.

In the future, we will evaluate the proposed approach by introducing other variants of VRP such as the Multi Depot Vehicle Routing Problem (MDVRP) and the Vehicle Routing Problem with Time Window (VRPTW). Moreover, the atomic VRP models can be implemented using other types of meta-heuristics and the combination of these models can be performed by defining other input and output connections.

**Author Contributions:** Conceptualization, S.S.G.; methodology, A.L.; software, M.H. and A.L.; validation, S.S.G. and A.L.; formal analysis, A.B.; investigation, A.E.; resources, M.A.; data curation, M.H.; writing—original draft preparation, S.S.G.; writing—review and editing, A.L. and M.H.; visualization, A.B. and A.E.; supervision, A.L.; project administration, A.L. and M.H.; funding acquisition, M.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

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

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.



**Acknowledgments:** We would like to express our gratitude to Yagoub Mohammed Amine for their meaningful support of this work.

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

## References

1. Pokrovskii, V.N. *Thermodynamics of Complex Systems*; IOP Publishing: Bristol, UK, 2020; pp. 2053–2563. [CrossRef]
2. Comin, C.H.; Peron, T.; Silva, F.N.; Amancio, D.R.; Rodrigues, F.A.; da F. Costa, L. Complex systems: Features, similarity and connectivity. *Phys. Rep.* **2020**, *861*, 1–41. [CrossRef]
3. Dorigo, M.; Floreano, D.; Gambardella, L.M.; Mondada, F.; Nolfi, S.; Baaboura, T.; Birattari, M.; Bonani, M.; Brambilla, M.; Brutschy, A.; et al. Swarmoid: A novel concept for the study of heterogeneous robotic swarms. *IEEE Robot. Autom. Mag.* **2013**, *20*, 60–71. [CrossRef]
4. Liboni, G. Complex Systems Co-Simulation with the CoSim20 Framework: For Efficient and Accurate Distributed Co-Simulations. Ph.D. Thesis, Université Côte d’Azur, Nice, France, 2021.
5. Camus, B.; Bourjot, C.; Chevrier, V. Considering a multi-level model as a society of interacting models: Application to a collective motion example. *J. Artif. Soc. Soc. Simul.* **2015**, *18*, 7. [CrossRef]
6. Nazari, M.; Oroojlooy, A.; Snyder, L.; Takác, M. Reinforcement learning for solving the vehicle routing problem. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 9861–9871.
7. Vidal, T.; Laporte, G.; Matl, P. A concise guide to existing and emerging vehicle routing problem variants. *Eur. J. Oper. Res.* **2020**, *286*, 401–416. [CrossRef]
8. Wang, Z.; Sheu, J.B. Vehicle routing problem with drones. *Transp. Res. Part B Methodol.* **2019**, *122*, 350–364. [CrossRef]
9. Li, Y.; Soleimani, H.; Zohal, M. An improved ant colony optimization algorithm for the multi-depot green vehicle routing problem with multiple objectives. *J. Clean. Prod.* **2019**, *227*, 1161–1172. [CrossRef]
10. Chen, W.; Ran, S.; Wu, C.; Jacobson, B. Explicit parallel co-simulation approach: Analysis and improved coupling method based on H-infinity synthesis. *Multibody Syst. Dyn.* **2021**, *52*, 255–279. [CrossRef]
11. Gomes, C.; Thule, C.; Broman, D.; Larsen, P.G.; Vangheluwe, H. Co-simulation: State of the art. *arXiv* **2017**, arXiv:1702.00686.
12. Pierce, K.; Gamble, C.; Golightly, D.; Palacin, R. Exploring Human Behaviour in Cyber-Physical Systems with Multi-modelling and Co-simulation. In Proceedings of the International Symposium on Formal Methods, Pasadena, CA, USA, 24–27 May 2019; pp. 237–253.
13. Siebert, J.; Ciarletta, L.; Chevrier, V. Agents and artefacts for multiple models co-evolution. Building complex system simulation as a set of interacting models. In Proceedings of the Autonomous Agents and Multiagent Systems-AAMAS 2010, Toronto, ON, Canada, 10–14 May 2010; pp. 509–516.
14. Paris, T.; Tan, A.; Chevrier, V.; Ciarletta, L. Study about decomposition and integration of continuous systems in discrete environment. In Proceedings of the 49th Annual Simulation Symposium (ANSS 2016): 2016 Spring Simulation Multi-Conference (SpringSim’16), Pasadena, CA, USA, 3–6 April 2016.
15. Paris, T.; Ciarletta, L.; Chevrier, V. Designing co-simulation with multi-agent tools: A case study with NetLogo. In *Multi-Agent Systems and Agreement Technologies*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 253–267.
16. Camus, B.; Paris, T.; Vaubourg, J.; Presse, Y.; Bourjot, C.; Ciarletta, L.; Chevrier, V. Co-simulation of cyber-physical systems using a DEVS wrapping strategy in the MECSYCO middleware. *Simulation* **2018**, *94*, 1099–1127. [CrossRef]
17. Silva, M.A.L.; de Souza, S.R.; Souza, M.J.F.; de Franca Filho, M.F. Hybrid metaheuristics and multi-agent systems for solving optimization problems: A review of frameworks and a comparative analysis. *Appl. Soft Comput.* **2018**, *71*, 433–459. [CrossRef]
18. Mguis, F.; Zidi, K.; Ghedira, K.; Borne, P. Distributed approach for vehicle routing problem in disaster case. *Ifac Proc. Vol.* **2012**, *45*, 353–359. [CrossRef]
19. Martin, S.; Ouelhadj, D.; Beullens, P.; Ozcan, E.; Juan, A.A.; Burke, E.K. A multi-agent based cooperative approach to scheduling and routing. *Eur. J. Oper. Res.* **2016**, *254*, 169–178. [CrossRef]
20. Silva, M.A.L.; de Souza, S.R.; Souza, M.J.F.; Bazzan, A.L.C. A reinforcement learning-based multi-agent framework applied for solving routing and scheduling problems. *Expert Syst. Appl.* **2019**, *131*, 148–171. [CrossRef]
21. Kim, T.G.; Praehofer, H.; Zeigler, B. Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex. Dynamic Systems. 2000. Available online: <https://www.semanticscholar.org/paper/Theory-of-Modeling-and-Simulation%3A-Integrating-and-Zeigler-Praehofer/92131a3aed0d72ccfe92364eee87e228c9c773e6> (accessed on 29 March 2022).
22. Van Tendeloo, Y.; Vangheluwe, H. Discrete event system specification modeling and simulation. In Proceedings of the 2018 Winter Simulation Conference (WSC), Gothenburg, Sweden, 9–12 December 2018; pp. 162–176.
23. Ricci, A.; Viroli, M.; Omicini, A. Give agents their artifacts: The A&A approach for engineering working environments in MAS. In Proceedings of the 6th international joint conference on Autonomous Agents and Multiagent Systems, Honolulu, HI, USA, 14–18 May 2007; pp. 1–3.
24. Camus, B.; Vaubourg, J.; Presse, Y.; Elvinger, V.; Paris, T.; Tan, A.; Chevrier, V.; Ciarletta, L.; Bourjot, C. Multi-Agent Environment for Complex Systems Cosimulation (Mecsyco)-Architecture Documentation. 2016. Available online: <http://mecsyco.com/dev/doc/User%20Guide.pdf> (accessed on 29 March 2022).

- 
25. Savelsbergh, M. *Vehicle Routing and Scheduling*; Technical Report; Georgia Tech Supply Chain and Logistics Institute: Atlanta, GA, USA, 2002.
  26. Omicini, A.; Ricci, A.; Viroli, M. Artifacts in the A&A meta-model for multi-agent systems. *Auton. Agents Multi-Agent Syst.* **2008**, *17*, 432–456.
  27. Ricci, A.; Piunti, M.; Viroli, M. Environment programming in multi-agent systems: An artifact-based perspective. *Auton. Agents Multi-Agent Syst.* **2011**, *23*, 158–192. [[CrossRef](#)]
  28. Gooding, T. Netlogo. In *Economics for a Fairer Society*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 37–43.
  29. Uchoa, E.; Pecin, D.; Pessoa, A.; Poggi, M.; Vidal, T.; Subramanian, A. New benchmark instances for the Capacitated Vehicle Routing Problem. *Eur. J. Oper. Res.* **2017**, *257*, 845–858. [[CrossRef](#)]