Routing and schedule simulation of a biomass energy supply chain through SimPy simulation package

T.M. Pinho
Universidade de Trás-os-Montes e Alto Douro, UTAD,
Escola de Ciências e Tecnologia, Vila Real, Portugal and
INESC TEC Technology and Science, Campus da FEUP, Porto, Portugal

J.P. Coelho
Instituto Politécnico de Bragança, Escola Superior de Tecnologia e Gestão,
Campus de Sta. Apolónia, Bragança, Portugal and
INESC TEC Technology and Science, Campus da FEUP, Porto, Portugal

P.M. Oliveira
Universidade de Trás-os-Montes e Alto Douro, UTAD,
Escola de Ciências e Tecnologia, Vila Real, Portugal and
INESC TEC Technology and Science, Campus da FEUP, Porto, Portugal

B. Oliveira and A. Marques
INESC TEC Technology and Science, Campus da FEUP, Porto, Portugal

J. Rasinmäki
Simosol Oy, Riihimäki, Finland

A.P. Moreira
Faculty of Engineering, University of Porto, Porto, Portugal and
INESC TEC Technology and Science, Campus da FEUP, Porto, Portugal

G. Veiga
INESC TEC Technology and Science, Campus da FEUP, Porto, Portugal, and

Publishers note: The publisher wishes to inform readers that the article "Routing and schedule simulation of a biomass energy supply chain through SimPy simulation package" was originally published by the previous publisher of *Applied Computing and Informatics* and the pagination of this article has been subsequently changed. There has been no change to the content of the article. This change was necessary for the journal to transition from the previous publisher to the new one. The publisher sincerely apologises for any inconvenience caused. To access and cite this article, please use Pinho, T. M., Coelho, J. P., Oliveira P. M., Oliveira, B., Marques, A., Rasinmmäki, J., Moreira, A. P., Veiga, G., Boaventura-Cunha, J. (2021), "Routing and schedule simulation of a biomass energy supply chain through SimPy simulation package", *Applied Computing and Informatics*. Vol. 17 No. 1, pp. 36-52. The original publication date for this paper was 30/06/2018.

J. Boaventura-Cunha

Universidade de Trás-os-Montes e Alto Douro, UTAD,
Escola de Ciências e Tecnologia, Vila Real, Portugal and
INESC TEC Technology and Science, Campus da FEUP, Porto, Portugal

Abstract
The optimisation of forest fuels supply chain involves several entities actors, and particularities. To successfully manage these supply chains, efficient tools must be devised with the ability to deal with stakeholders dynamic interactions and to optimize the supply chain performance as a whole while being stable and robust, even in the presence of uncertainties. This work proposes a framework to coordinate different planning levels and event-based models to manage the forest-based supply chain. In particular, with the new methodology, the resilience and flexibility of the biomass supply chain is increased through a closed-loop system based on the system forecasts provided by a discrete-event model. The developed event-based predictive model will be described in detail, explaining its link with the remaining elements. The implemented models and their links within the proposed framework are presented in a case study in Finland and results are shown to illustrate the advantage of the proposed architecture.

Keywords Biomass supply chain, Discrete-event simulation, Operational planning

Paper type Original Article

1. Introduction
Supply chains are complex systems that involve several partners and intensive flows of material, information and financial issues [1–3]. With the current competitiveness, companies must be able to timely comply with customers' demands, assuring the products quality, while constantly being aware of their profits [4,5]. In this sense, an efficient material flow management mechanism will have a positive impact regarding all these issues [6]. Furthermore, resilience to external disturbances should be an essential characteristic of any supply chain. That is, when affected by unexpected events, it must have the ability to react, recover and overcome the effects of these phenomena [7]. In a broad sense, supply chain management aims to increase the performance of the overall supply chain, improving responsiveness, efficiency and profitability of the involved stakeholders [4,8]. This supply chain management quality is tightly tied to the knowledge about the system behaviour [3]. As more the system reaction is understood, the better the decisions that are taken. Although, due to the intensive interaction and interdependency between the supply chain players, it becomes hard to grasp the involved dynamics by means of simple inference or analytic methods [8]. Computer-based simulation, in particular discrete-event models, is a commonly used method to represent systems not restricted by rigid mathematical structures [9–11]. It is a powerful tool to test and reproduce different scenarios and forecast the impact of distinct configurations in the supply chain design and behaviour [12]. By means of computer-based simulation, it is possible to dynamically change the variables within the supply chain [5,8,13], allowing evaluation of the system's reaction to changes, while incorporating uncertainty over different operating scenarios [9].

This paper addresses a case study of a biomass supply chain located in Finland. Biomass represents an important part in a new energetic paradigm based on renewable sources [13–19]. Currently, it accounts for approximately 10% of the total global primary energy consumption [20]. The explanation for this fact can be tracked down to several advantages, namely its versatility and storage capability. The former is due to biomass's ability to be used to produce electricity, heat and biofuels. The latter is self-explanatory since biomass can be stored in bulk, allowing energy generation on-demand [21–24]. The biomass supply chain is a very complex process since it deals with several nodes and requires synchronization between machines, clients and transportation. For these reasons, this kind of processes involves long time strategic

planning, short term resources allocation and process disturbances mitigation. If no proper decision support tools are used, even for small scale setups, this degree of complexity often leads to intractability. In this sense, this paper proposes a new methodology to increase the resilience and flexibility of a biomass supply chain through a closed-loop control system based on the system forecasts provided by a discrete-event model, at the operational level. The proposed structure aims to evaluate the impact of unexpected events, such as delays and failures, in the overall supply chain performance and to take actions if deviations from a set-point schedule plan are considered relevant.

The next section will enumerate the typical problems faced when trying to control this supply chain, while targeting the highest possible economic income, flexibility and resilience. Section 3 will detail a new strategy that can be used to mitigate the set of problems previously enumerated. Section 4 will present the event-based model devised to integrate this strategy, describing, in Section 5, the flag generator system, and in Section 6, a practical example and the results achieved. Finally, the main contributions and suggestions for future work are addressed in Section 7.

2. Problem statement

More than 75% of Finland's surface area is covered by forest. Taking advantage of this condition, proper legislation and meticulous reforestation policies, biomass plays several important roles within the socio-economic structure of Finland [25]. It promotes economy in rural areas and is a key product in both the pulp/paper industry and in co-generation power plants. Regarding the latter, Finland holds first place among the current International Energy Agency country members, in terms of biomass percentage in the overall primary energy supply [26]. Currently, more than 20% of its total primary energy supply is originated from solid biomass leading to more than 7.5 million tonnes of oil equivalent fuel.

The Finnish goals for the year 2020, in terms of energy consumption based on renewable energy sources, are even more ambitious, which will lead to an increase in production and consumption of wood chips. Furthermore, improvements in both production rates and competitive price, compared to other energetic sources, should be achieved. In this sense, to attain all these objectives, the wood chips biomass supply chain should be carefully managed and monitored. This close supervision will be a key factor to predict and prevent potential supply chain failures, while maintaining efficient resources allocation.

The usual biomass supply chain network comprises a high number of wood piles and a given number of clients, in our case power plants, scattered along a given region. Those clients require a certain energy amount to be delivered every week. The wood piles, i.e., timber harvest residue piles composed by branches and tree tops, placed at the roadside, are chipped and the resultant product is transported to the power plants by means of a trucks fleet. This concept is illustrated in Figure 1. It is worth noting that, in the present case study, both chippers and trucks start and end their working day at a specific network node denoted as depot.

The planning of this supply chain involves two distinct decision levels: a tactical and an operational level. The tactical planning level generates decisions regarding the daily material assignments and fleet dimensioning over a period of a week. Based on the defined assignments, the operational planning level defines, for each working day, vehicles routing and schedules [27–31].

However, the conditions considered by both tactical and operational planning levels do not remain constant during the operation time horizon originally established, since machine breakdowns or changes may occur in operating conditions, among other factors. Hence, there is frequently the need to derive a new working plan to circumvent those problems, while promoting business efficiency. Depending on the problem severity, this re-scheduling process
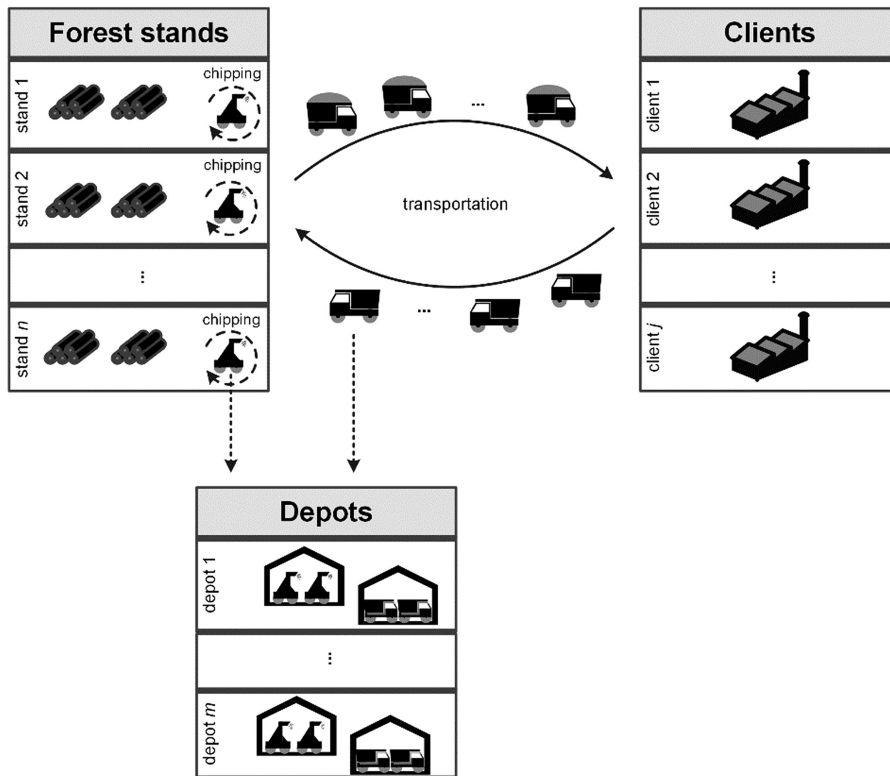
may lead to considerable changes in the original working plan. Those changes can cause
several logistic problems and may not even be practicable due to the system inertia. For this
reason, this task must be carried out parsimoniously and only when there is strong evidence
that the current system state will lead, at the end of the working day, to an outcome that is far
from the planning target for that day. This condition requires the evaluation of the impact on
the business performance due to machines failure or other sources of disturbance, such as
delays due to weather conditions or traffic. This evaluation must be made through system
forecasts and data analysis. Due to the biomass supply chain complexity, the prediction of
future outcomes derived from changes in the current system states cannot be made easily
without a dynamic process model. With such a model, it is possible to obtain a good estimate
on the impact of disturbances in the final working day regarding operating costs and clients
supply faults. By implementing this model in a computer, it is possible to access this data in
real-time and make decisions on-the-fly regarding eventual changes in the working plan. It is
this closed loop control strategy that is addressed in this paper. Namely, a new paradigm is
presented for both control and decision support directed to biomass supply chains
operational level. The present approach uses a discrete event-based model of the supply chain
in order to generate forecasts when deviations of the system states are observed. These
predictions, when compared to the goal defined by the planning team schedule, will be used to
feed a classification system that takes decisions about the need of replanning. In this
framework, a new schedule is required if the forecast deviations are considered severe enough

to justify the generation of a new plan. This is due to the high computational load associated with the optimisation algorithm, as well as the time and resources required to the plan implementation. The following section will further explore this paradigm by presenting the details regarding the controller architecture.

3. Planning and control system architecture

To minimize the disturbances impact in biomass supply chains, a new planning and control system architecture is proposed. The supervision and control methods were developed under the FOCUS (Advances in Forestry Control and aUtomation Systems in Europe) FP7 European project and applied to a wood chips production and delivery company located in Finland. In Figure 2, a block diagram that represents the main operations within the devised planning and control system architecture is presented. In this work, specific detail is provided regarding the shadowed area of Figure 2.

The closed-loop system structure comprises an operational planning module, a remote computer server, an event-based predictive model block and a flag generator system. The remote server is a hardware device where databases are stored. The information stored in those databases regards several variables such as system states and flags (see Section 5). Those databases can be remotely accessed and modified by other network nodes, including the partners responsible for the supervision, management and planning of the supply chain.

The processing loop is initiated at the beginning of the week when the power plants provide, to the tactical planning level, the amount of energy to be delivered to each of them at the end of a week. Based on this energy target, and by also considering the network description and its resources, such as available chippers, trucks and wood piles, the tactical planning solves a mixed integer programming (MIP) problem that, if a feasible solution is found, will lead to a daily assignment that will ensure, at the end of the week, the supply contracts [32]. Although an important level, the tactical decision layer is not considered in detail in the present work.

This daily service assignment is communicated to the operational planning level, through the remote server, which establishes the daily routes and schedules throughout matheuristic methods.

Namely, the operational planning module includes a decomposition-based solution method encompassing clustering and an exact MIP model for the daily synchronized routing and scheduling of chippers and trucks. The solution method starts by clustering services able to be visited together, based on roadside comminution operational rules and on proximity-based measures [33] between different sets of services. Next, for each of the resulting service clusters, a MIP model for the synchronized routing of chippers and trucks is applied, a variant
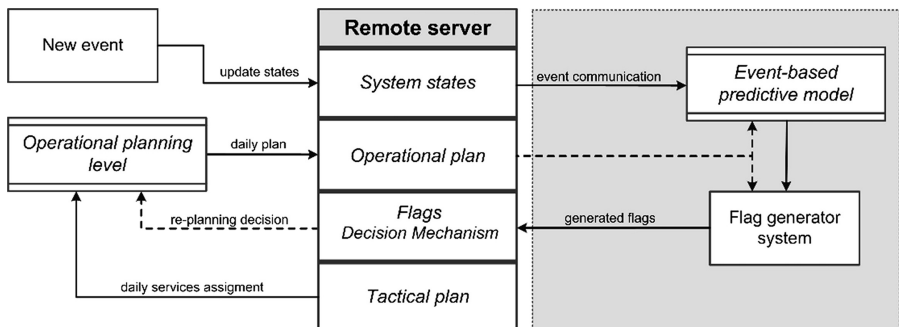

Figure 2.
Supervision, planning and control system architecture.

of the Vehicle Routing Problem with Multiple Synchronization constraints (VRPMS) [34–36], where the objective is to minimize the used chipper and truck fleet, transportation costs and unproductive working times while accounting for synchronization constraints between vehicles. The operational planning module solution method has been implemented using Java integrated with GUROBI 6.0.5 optimization software used to solve the routing and scheduling MIP model. The output of the optimization module provides the daily schedules of chipper and truck at the terminals and piles, which is sent by the server to the personal and operators on the field.

During the day, events are communicated from the field workers to the remote server leading to system state updates. These events regard the entering and exiting of machinery in piles, depots or clients. Moreover, those events can describe system anomalies such as machine malfunctions. When the system states are updated, the remote server notifies the event-based predictive model. It then forecasts the system behaviour toward the end of the working day by populating its internal states with the information derived from the current list of states provided by the remote server. In order to foresee the effects of each communicated event in the final delivery services, a computer model of the supply chain was built. Due to the system's discrete and aperiodic nature, an event-based model was formulated. In particular, the devised model was developed using the software toolbox SimPy [37]. The SimPy is a discrete-event framework that runs over the Python programming language. In general, SimPy toolbox allows the modelling of asynchronous tasks by means of a set of primitive operations such as events, processes and resources. These primitives will be used for modelling the distinct components that encompass the forest-based supply chain, such as machines, clients and vehicle routing. A detailed description about the model formulation will be presented in the following section.

Based on the predicted outputs computed by the event-based model, and by comparing their results with the target imposed by the current working plan, the flag generator system will gather the information that will be used by the replanning decision block. If replanning is required, the operational planning level is invoked, and a new plan is generated. When this re-vplanning is triggered, information about the state of the system in the service cluster at the time of the disruption is passed on to the operational planning module. This information encompasses the available vehicles, their location and the remaining services to be completed of the disrupted service cluster. This information is used in a pre-processing stage where a tailored graph is constructed reflecting these characteristics. Since some vehicles may be completing services at the time disruption occurs, one simply establishes that the initial time and location of vehicles corresponds to the time and location where their on-going services would end. This procedure is used to ensure that vehicles are only able to be re-planned after they have completed their on-going service. After the pre-processing stage, the MIP model used in the operational planning module is used to re-compute vehicles' routes and schedules for the disrupted cluster. By considering solely the disrupted cluster and performing the pre-processing of the graph, it is aimed to address the need for fast solution computational time, as required by daily re-planning and dispatching.

This new plan is communicated to all the pertinent supply chain players and the loop continues as previously described with the arriving of new events. In the remaining sections, further insights on the event-based predictive model and on the flag generator system will be provided.

4. Event-based model formulation
This section deals with the computer implementation of the supply chain model by means of its description within an event-based software paradigm. In this work, the SimPy library was used to emulate the behaviour of each supply chain object as well as its interdependencies. In

this computer modelling language, each one of the physical elements is represented by means of a Python class. In particular, any chipper, truck, wood pile, client and depot represented in Figure 1, is instantiated as an object of its respective class data type. In addition, the set of admissible behaviours of each supply chain element is described as methods within the corresponding class.

In this section, each of the distinct classes is documented. That is, the classes structure is presented, but the methods description is depleted of unnecessary programming details.

Regarding the model operating sequence, and as can be observed from Figure 2, the simulation starts whenever a new event occurs. The model progression along its states is illustrated in the flowchart present in Figure 3. The first step consists of checking the operability of all the machines involved in the present working day.

Whenever a machine is unable to perform its tasks, it is forwarded to a "breakdown" state, where it will remain until the end of the simulation. On the other hand, if it is operational, the daily working plan is accessed from the remote server and delivered to the working machine. As defined in Section 2, the chippers can only move between piles, while trucks can travel between piles and clients. The assigned tasks for each machine are sequentially executed until the task's daily list ends. When the last task is attained, the machines are forwarded to the "depot", where they remain until the end of the simulation. Notice that both the "breakdown" and "depot" states are defined, within the Python computational environment, as classes. Besides the situations of breakdown and depot permanence in non-working
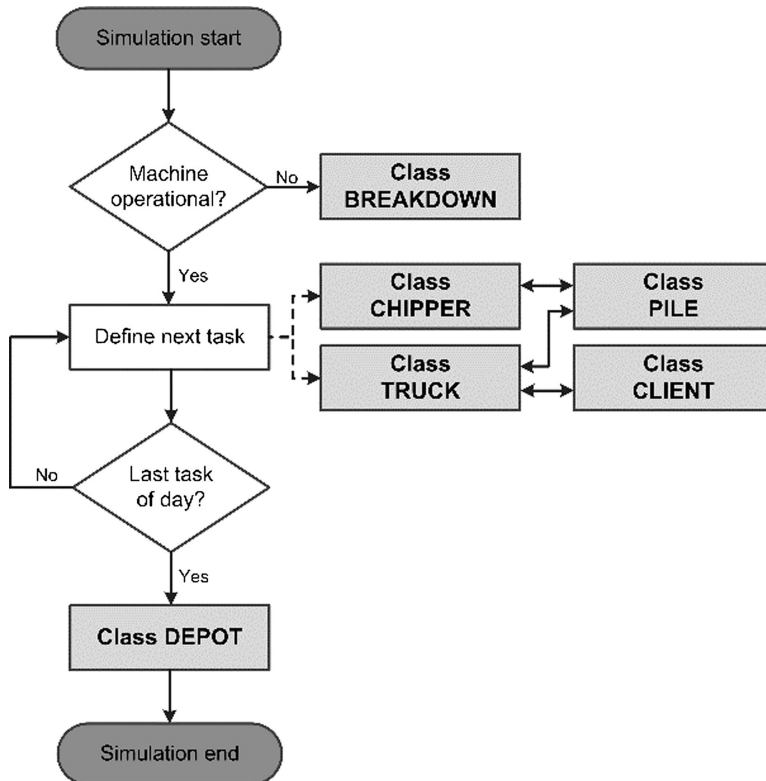

Figure 3.
SimPy model
flowchart.

periods, the truck can also be interrupted during its loading process due to schedules obligations. This forced situation is computationally handled by issuing a service interruption through a dedicated Python class designated by "interrupted". This approach means that, if a truck arrives with a delay to a loading task, the chipper does not change its schedule due to this truck lag and will leave the wood pile according to its schedule. Thus, if the chipper schedule ends before completing the loading task, the charging process is interrupted, and the truck is dispatched to the "interrupted" class, where it will remain until the end of the simulation. This behaviour means that the model, by itself, only reacts to the actual system states and does not take any decision. The unloading process can also send the trucks to the interruption state. As a matter of fact, the "client" can also be responsible for a truck object interruption due to the unloading time-windows. This situation is detailed ahead.

Up to now it was established that there are many players within the biomass supply chain. Each protagonist is computationally characterized by means of classes data types populated with its methods. The "chipper" class allows each chipping machine to execute sequentially the planned tasks for a full working day. Besides the attributes specification, this class is composed by five main methods. The first method assesses the machine operability using the event table that has triggered the simulation. If a breakdown is detected, no other class method is called, and the chipper is dispatched to the "breakdown" class. Otherwise, the method concerning the plan reading is accessed and the next queued task is retrieved. The chipping machine displacement process is carried out by the "imdriving( )" method. The driving task can be viewed as a time-delay whose value is equal to the time elapsed since the chipper departure up to its arrival at the destination point. In SimPy, this behaviour can be modelled through a timeout event. When a chipper reaches a destination wood pile, a flag is issued to the "pile" class. This information interchange between the chipper and the pile will be used for synchronization purposes between chippers and trucks during the loading process. In a similar way, when the chipper leaves a wood pile, the "pile" class is informed. Without the presence of a chipper in a wood pile, the chipping and truck loading processes are impossible.

The last three methods described that regards the chipper current state are called from within the "imreadingplan( )" class method. The "imdriving( )" and "imatpile( )" methods are sequentially executed until the end of task list. At last, the method "imatdepot( )" is called and the chipper is dispatched to the depot.

The "truck" class is similar to the "chipper" class. If fully operational, the truck working plan is read from the schedule and the driving, loading and unloading tasks are executed sequentially until the depot task is attained. In order to carry out the loading service, when a truck reaches a pile, it communicates its arrival and makes a request to the pile to be synchronized with the chipper. This request is performed by means of a yield command, thus suspending the truck process until the yielded event is executed.

In parallel, the "pile"'s class synchronization method is triggered. The "sync( )" method will verify the presence of a chipper in the pile and, if positive, it will give permission to the truck to start loading. It should be noticed that, even if the loading process is still running, when the schedule plan indicates that the chipper should end that current task, the "pile" class is informed about it and will react by interrupting the truck loading process. If this is the case, the truck will be forwarded to the "interrupted" class where it will stand until the end of the simulation. This reaction can lead to a severe impact in the number of programmed tasks attained at the end of the working day. This is the case due to the remaining tasks attributed to the interrupted truck will be overlooked leading to the loss of accomplishment of several services.

On the other hand, if the loading service is successfully accomplished, the truck object will continue reading its schedule plan and retrieve the next task. The task following the loading is always an unloading at some power plant client. This means that, after loading, the truck

will drive towards a new target client. This activity is modelled by means of a timeout whose value is equal to the ratio of the distance between the wood pile and the client and the truck's speed. After this timeout, an unloading permission request is issued by the truck to the client object. The response to this request will depend on the client unloading time window. That is, each client has specific time windows during which it is permitted for the trucks to unload. Based on this window's lower and upper limits, a truck can be delayed or even interrupted.

Regarding the "depot", "breakdown" and "interrupted" classes, all have a similar structure. The class constructor initializes the object with the total simulation time and a set of descriptors related to it. Those descriptors may include an identifier, and the list of trucks or chippers attached to it. The machines to be added to those classes are stored in "FilterStores". In the SimPy framework, "FilterStores" inherit from "Stores" which refer to shared resources for storing a given amount of objects with the same characteristics. In particular, for "FilterStores", the stored objects can be differentiable through particular characteristics, such as, an identification label. Furthermore, all the classes identified have a method designated by "whosinclass( )", called at the end of the simulation, that retrieves the identification of the machines present in that same class.

The interconnection between all the above mentioned classes can be made clearer by observing the diagram presented in Figure 4. This diagram exhibits, in detail, the dependence and the interactions between the involved network objects. A set of such objects, according to the network structure and scheduled machines, must be instantiated. All those objects have their existence within a special environment formalized by the SimPy framework. This environment supports and keeps track of the occurring events and will also be responsible for the simulation time tracking. In particular, the chippers, trucks, piles, clients and depots are embedded into Python dictionaries of objects. Once all those elements are created, the simulation can be started. After completed, the simulation results will be used in a classification system that will determine whether or not a replanning is needed, as described in Section 3.
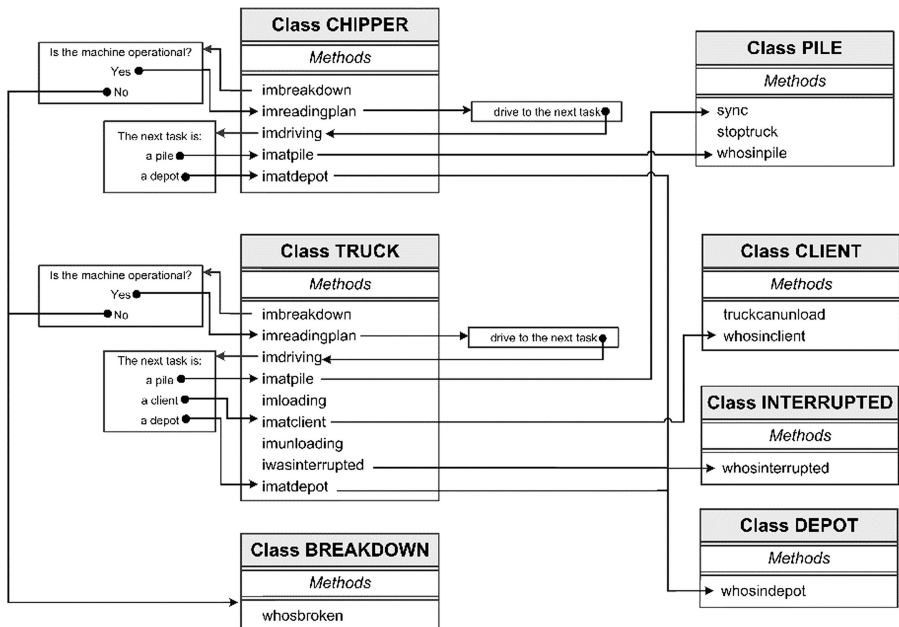

Figure 4.
Interaction between the model classes.

This classification system will have as input features the deviations in time and in accomplished services, regarding the initial plan delivered from the planning level. The classification system output is described in the form of a set of flags. Those flags are submitted to a supervisor system that will decide, based upon these, if a replanning is required or not. The following section will be devoted to further explain how this decision process takes place.

5. Flag generation system

For a typical size biomass supply chain, the planning process is a very complex task. If the system was described as a mixed integer programming problem, the dimension imposed by the number of network nodes and constraints would make it difficult to be solved in a reduced time scale. Hence, the need of replanning a working day must be carefully decided. It is not just a question of the computational costs, but also about the logistics involved to reformat the human resources work schedule in the field. The decision to replan should depend on the severity of the system state changes regarding the ones initially assumed. For example, in case of a chipper breakdown, what is the impact that this event will have in the final deliveries? This projection is not an easy task to be made by a human when several variables are in stake. In this background, the model described in the previous sections can be used to easily perform this type of forecasts. Nevertheless, it is also necessary to interpret the results provided by the model and decide whether a new replanning is really necessary. This task will be carried out by a classification system that takes the model simulation results and provides a set of flags that, in conjunction, describes the mismatch severity between the prediction outcome and the original schedule plan. This section explains how those flags are generated and their purpose. The flag generation system will be called by the software after the supply chain simulation finishes. This system is responsible for generating a hierarchical flag structure that will be sent to a remote server detailing the mismatches between the expected results generated by the model simulation and the original operational plan according to the recent system states. Based on that information, the planning team can reschedule the tasks in order to react beforehand and, thus, prevent the predicted deviations. For each planned task, the deviations between the predicted and initially assumed start and end times are computed. Those deviations are represented in Figure 5 and marked with numbers spanning from 1 to 4.

Additionally, the number of predicted unloading tasks at the clients is compared with the ones presented in the initial plan. This difference leads to the computation of deviations type 5 and 6 as follows. Let τ_i^s be the planned start time of task i and $\widehat{\tau}_i^s$ be its prediction. The deviation δ_i^s between them is given by

$$\delta_i^s = \tau_i^s - \widehat{\tau}_i^s \tag{1}$$

If δ_i^s is greater or equal than a threshold value x_s, a type 1 deviation will occur. On the other hand, if it is smaller or equal to a threshold value $-x_s$, then one will have a deviation type 2. In the same context, let τ_i^e be the planned end time of task i and $\widehat{\tau}_i^e$ be its prediction. Then,

$$\delta_i^e = \tau_i^e - \widehat{\tau}_i^e \tag{2}$$

regards the end time deviation error. Following the same as in the previous case, if $\delta_i^e \geq x_e$ a type 3 deviation is considered and if $\delta_i^e \leq x_e$ the deviation is of type 4. The generated flags will depend on the cumulative value of the absolute deviations denoted by Δ_t. This cumulative value is computed as:

$$\Delta_t = \sum_{i \in I} \left| \delta_i^s \right| + \left| \delta_i^e \right| \tag{3}$$
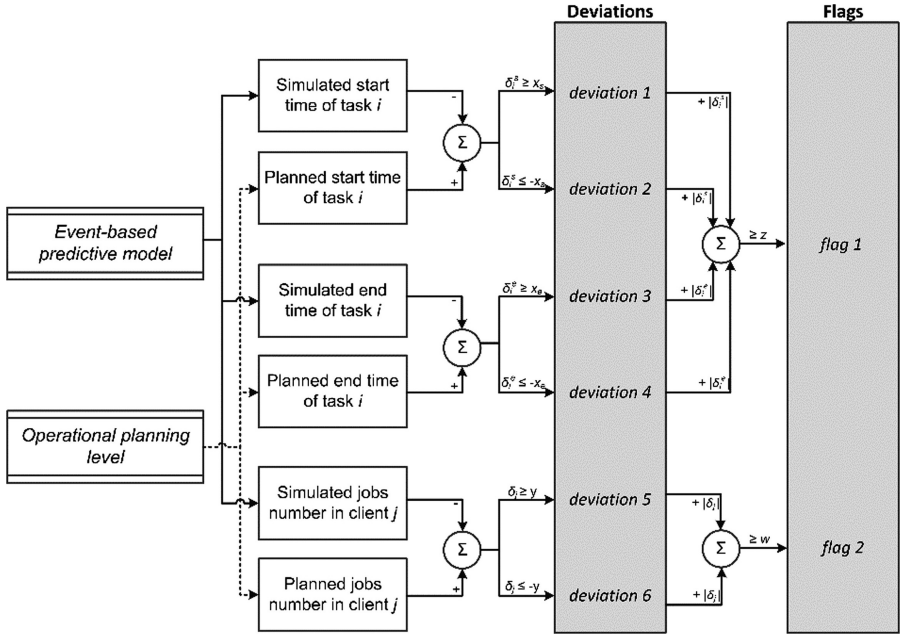
Figure 5.
Structure of the flags
generation system.

where I regards the set of all the tasks in the present schedule. If Δ_t is larger, or equal, to a predefined value z then a type 1 flag is generated. As already noticed, besides the time, a deviation in the initial proposed number of jobs is also important in the replanning decision process and can lead to a type 2 flag. In order to this flag being issued, the condition $\Delta_c \geq w$ must be met with,

$$\Delta_c = \sum_{j \in J} |\delta_j| \tag{4}$$

as the sum of the absolute deviations regarding the planned job deliveries at the target clients set J, and w is just a threshold value. Observe that now,

$$\delta_j = J_j - \widehat{J}_j \tag{5}$$

where J_j regards the number of planned deliveries at client j and $\widehat{J}_j$ is its predicted value using the event-based model. To conclude, it can be stated that the sum of the deviations' absolute values give rise to two flag types. Differences at time level lead to a type 1 flag, and mismatches regarding the number of unloading services will lead to a type 2 flag. The result of this classification process is embedded into a JSON markup data file and sent to the remote server where it will be used to decide whether a new plan needs to be generated.

In order to illustrate how this procedure will work in closed loop, the next section is devoted to the presentation of a practical example. A given biomass supply chain is defined and the effect of distinct network nodes faults will be analysed. In particular, the results concerning the effect of machines malfunctions or time delays will be shown.

6. Practical example

To demonstrate the applicability of the described methodology, an instance will be used as an example. The biomass supply chain considered for this practical example encompasses a set of five piles, five power plants, four trucks, two chippers and a total of 20 services (10 loading and 10 unloading services). The presented practical example is a part of the entire plan produced by the solution method for the operational synchronization of chipping and delivery. Only a part of the problem is presented due to simplicity in the description and space restrictions. The solution method decomposes the entire problem into clusters, each encompassing a subset of all loading and unloading services for a given day. If a disruption occurs, only the affected cluster is analysed. This allows focusing the simulation and the replanning on a small subset of services, which are the most relevant for the synchronization between both chippers and trucks. The proposed methodology allows to solve the re-planning problem very fast, an essential requirement when dealing with near real time control of operations. Aside from this, one also aims towards the minimization of abrupt changes in the plan, that is, one keeps the assignment of vehicles to locations so that drivers' schedules suffer the least changes possible. Only the order in which the different services are performed and the schedules of the vehicles are re-planned. The re-optimisation module uses a similar MIP model developed for the joint routing and scheduling of chippers and trucks, which has been modelled using GUROBI JAVA API [38].

In Figure 6, a daily plan is depicted of the above referred instance proposed by the operational planning level. Each task is represented by a start time and an end time (enhanced with a red flag) and the respective pile (p) or client power plant (c) number. At the end of the day, each equipment has a task associated to the depot (d) where it will remain until the following working day. For instance, truck 13 starts its daily tasks with a first unloading service at pile 6 (p6) at time unit (t.u.) 13. The service has a duration of 45 time units and, as such, that task ends at time 58. It then drives to the following destination, in this case client 6 (c6), in which it should arrive at time unit 86. As previously mentioned chippers move between piles within a working day, trucks between piles and clients and both equipments end the day in the depot.

Concerning the developed planning and control architecture, and in particular the event-based predictive model devised to forecast the impact of events in the working plan, this practical example will be used to estimate the dynamic system behaviour regarding different operating scenarios. It is important to mention that the operational plan developed has embedded buffer time, represented in Figure 6 with the dotted lines, that enables to attenuate the effect of delays in some tasks. In this sense, the occurred deviations from the plan can be suppressed and the equilibrium re-established, without the need of a full reschedule. The proposed example is used to study a particular type of disruption, namely, let's consider a delay of ten time units in truck 13 in its first daily task due to adverse weather conditions. This delay event is communicated by the remote server to the event-based predictive model in order to assess its impact in the remaining work plan. Based on the plan depicted in Figure 5, the event-based predictive model will instantiate the required elements. In detail, the equipment (chippers and trucks) used, and the piles, clients and depots visited. The event communication will contain the last known position and respective time of every equipment, as well as its operational state. With this information, as described in Section 4, the model will be able to simulate the impact of the event and forecast possible deviations from the plan. With this methodology, one can react beforehand and prevent the predicted deviations, by means of flags and schedules adjustments whenever necessary. As a matter of fact, in the current real systems, no monitoring and corrective actions are performed and, as a consequence, minor delays that can have severe impacts in the course of the working day are not considered. Thus, taking into consideration the highly dynamic and interdependence of tasks in the current supply
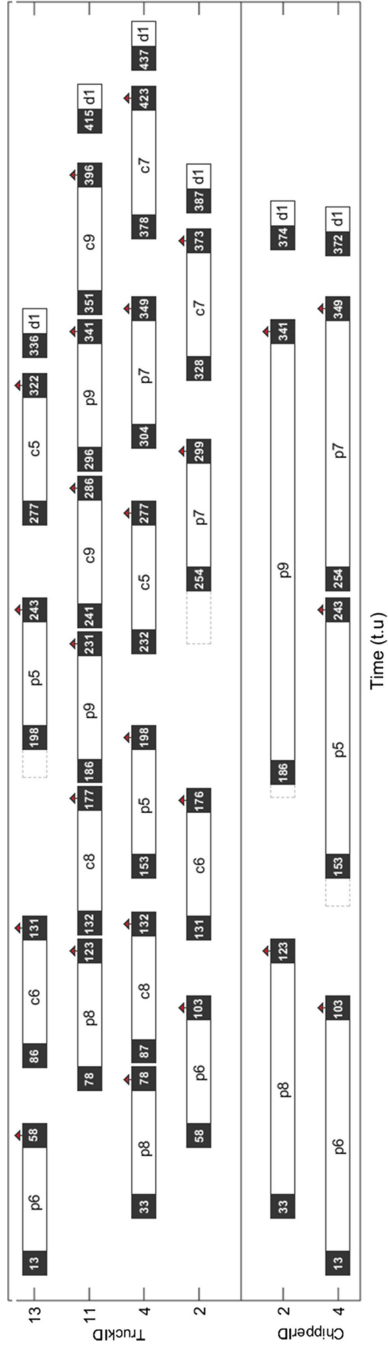
Figure 6.
Daily schedule plan
defined by the
planning department
(p – pile, c – client,
d – depot).

chain, the proposed methodology can be used as a fast and efficient management tool able to monitor and control the system's actions. Furthermore, the model can be also used in a first instance to analyse the robustness of the developed plan, for example by introducing stochasticity to the driving times.

Following the simulation performed by the event-based predictive model that will forecast the tasks accomplishment based on the newly communicated event, as described a list of deviations and flags is generated through the flag generation system. Table 1 summarizes the obtained hierarchical alerts resultant from the initial delay in truck 13 (t13).

Regarding the introduced disturbance, one is able to notice that a small delay can have relevant impacts in the tasks accomplishment. Particularly with this, besides the delays propagated through the remaining working period in the own delayed truck and in other (t2), also, two unloading tasks will lack. As mentioned before, some trucks, and in particular truck 13, have buffer times before some tasks. This enables the dissipation of the delay in all its daily tasks, and consequently prevents impacts in other trucks schedules. It should be noted that the considered mismatch threshold for the time was 5 t.u. and for the unloading services was 1 service, in both deviations and flags. In this sense, based on Section 5, and in particular Figure 5, it can be observed that two type 2 and two type 4 deviations were generated. This means that the vehicles in which these deviations occurred had a delay (in this case of 10 t.u.) in the tasks' start and end times, respectively. Also, when the original plan and the projections performed by the event-based predictive model were compared regarding the number of unloading tasks, and thus the number of deliveries in the clients, the δ_j obtained was higher than the imposed threshold generating a deviation warning. As denoted by Eqs. (3) and (4), the flags are generated based on the cumulative value of the absolute deviations' amount. Considering that deviations concerning both time and unloading services are created, the two type flags are produced. In particular, as four deviations of -10 t.u. were generated, adding their absolute values, produce a total flag of 40 t.u.. Regarding the differences in the unloading services, as just one deviation was generated, the flag will assume its same value. Following, as denoted in Figure 2, the resultant flag file is delivered to the remote server, which will decide on the operational replanning possibility. In the particular case study presented in this work, it can be observed that unloading tasks will be unaccomplished. As a consequence, and because the event occurred at the beginning of the day, the prediction deviations can be recovered in the same day. For that, the operational planning level will be called in order to generate a new plan considering the event and its respective impacts. The newly generated plan is updated to the remote server and the event-based predictive model executes its future simulations based on the same. In the normal practice in the field, these tracking/preventive actions are not taken leading to higher costs and difficulties in the companies operation. The proposed approach constitutes an advantage that allows to reduce these impacts and provide more flexibility and efficiency to the system.

Alert type	Type	Service ID	Vehicle ID	Amount of deviation
Deviations	2	2	t13	-10 t.u.
	2	1	t2	-10 t.u.
	4	1	t13	-10 t.u.
	4	2	t13	-10 t.u.
	5	–	t2	2 unloading services
Flags	1	–	–	40 t.u.
	2	–	–	2 unloading services

Table 1.
Deviations and flags generated by the flag generation system by the event-based predictive model simulation (t.u. – time units).

7. Conclusions

Supply chains are complex systems that involve several entities with a strong interdependence. In this sense, and mainly in systems with considerable dimensions where the stakeholders are scattered along sparse regions, it becomes very difficult to ensure proper working of supply chain processes in the long run. This is due to several disturbances and changes that happens during the working day that often invalidates the assumptions made in the initial working plan. In order to overcome this issue, this work presented a novel adaptive control system mechanism. With this methodology it is possible to decide in a fast, informed and efficient way about the replanning necessity in the biomass supply chain processes. This control structure regulates the system behaviour by means of information flags transmitted to the planning level at discrete, and irregular, time instants. The information content on those flags depends on the supply chain prediction results obtained by an event-based model. Simulation results show that this controller paradigm can, in fact, be used in practical applications since it has lower computational load and lead to coherent plans whenever reference changes are required. The proposed approach constitutes an advantage face to currently used methodology which does not implement any monitoring/preventive actions. In the future, the proposed framework will be improved by including system's stochastic behaviours such as the influence of weather conditions and traffic constraints in driving times.

References

[1] F. Longo, G. Mirabelli, An advanced supply chain management tool based on modeling and simulation, Comput. Ind. Eng. 54 (3) (2008) 570–588.

[2] K. Subramanian, J.B. Rawlings, C.T. Maravelias, J. Flores-Cerrillo, L. Megan, Integration of control theory and scheduling methods for supply chain management, Comput. Chem. Eng. 51 (2013) 4–20.

[3] L.B. Janamanchi, J.R. Burns, Control theory concepts applied to retail supply chain: a system dynamics modeling environment study, Hindawi Publishing Corporation – Modell. Simul. Eng. 2013 (15) (2013) 1–14.

[4] C.M. Badole, R. Jain, A.P.S. Rathore, B. Nepal, Research and opportunities in supply chain modeling: a review, Int. J Sup. Chain. Mgt 1 (3) (2012) 63–86.

[5] B.-C. Park, S. Jeong, A modeling framework of supply chain simulation, J. Supply Chain Oper. Manag. 12 (2) (2014) 91–106.

[6] F. Persson, J. Olhager, Performance simulation of supply chain designs, Int. J. Prod. Economics 77 (3) (2002) 231–245.

[7] H. Carvalho, A.P. Barroso, V.H. Machado, S. Azevedo, V. Cruz-Machado, Supply chain redesign for resilience using simulation, Comput. Ind. Eng. 62 (1) (2012) 329–341.

[8] Y.H. Lee, M.K. Cho, S.J. Kim, Y.B. Kim, Supply chain simulation with discrete-continuous combined modelling, Comput. Ind. Eng. 43 (1) (2002) 375–392.

[9] S. Buckley, C. An, Supply Chain Management on Demand – Supply Chain Simulation, Springer, Berlin Heidelberg, The Netherlands, 2005.

[10] A.A. Tako, S. Robinson, The application of discrete event simulation and system dynamics in the logistics and supply chain context, Decis. Support Syst. 52 (4) (2012) 802–815.

[11] R. Cigolini, M. Pero, T. Rossi, A. Sianesi, Linking supply chain configuration to supply chain performance: A discrete event simulation model, Simul. Modell. Pract. Theory 40 (2014) 1–11.

[12] S. Terzi, S. Cavalieri, Simulation in the supply chain context: a survey, Comput. Ind. 53 (1) (2004) 3–16.

[13] M. Mobini, T. Sowlati, S. Sokhansanj, Forest biomass supply logistics for a power plant using the discrete-event simulation approach, Appl. Energy 88 (4) (2011) 1241–1250.

[14] F. Frombo, R. Minciardi, M. Robba, R. Sacile, A decision support system for planning biomass based energy production, Energy 34 (3) (2009) 362–369.

[15] A. Asikainen, Simulation of stump crushing and truck transport of chips, Scand. J. For. Res. 25 (3) (2010) 245–250.

[16] V. Buytaert, B. Muys, N. Devriendt, L. Pelkmans, J. Kretzschmar, R. Samson, Towards integrated sustainability assessment for energetic use of biomass: a state of the art evaluation of assessment tools, Renewable Sustainable Energy Rev. 15 (8) (2011) 3918–3933.

[17] L. Chen, X. Li, W. Wen, J. Jia, G. Li, F. Deng, The status, predicament and countermeasures of biomass secondary energy production in China, Renewable Sustainable Energy Rev. 16 (8) (2012) 6212–6219.

[18] U.J. Wolfsmayr, P. Rauch, The primary forest fuel supply chain: a literature review, Biomass Bioenergy 60 (2014) 203–221.

[19] T.R. Teixeira, C.A.A.S. Ribeiro, A.R. Santos, G.E. Marcatti, A.S. Lorenzon, N.L.M. Castro, G.F. Domingues, H.G. Leite, S.J.M.C. Menezes, P.H.S. Mota, L.A.A. Telles, R.S. Vieira, Forest biomass power plant installation scenarios, Biomass Bioenergy 108 (2018) 35–47.

[20] A. Karhunen, T. Ranta, J. Heinim, E. Alankangas, Market of Biomass Fuels in Finland an Overview 2013, Lappeeranta University of Technology, IEA Bioenergy, 2014.

[21] A. Rentizelas, A. Tolis, I. Tatsiopoulos, Logistics issues of biomass: the storage problem and the multi-biomass supply chain, Renewable Sustainable Energy Rev. 13 (4) (2009) 887–894.

[22] A.D. Meyer, D. Cattrysse, J. Rasinmäki, J.V. Orshoven, Methods to optimise the design and management of biomass-for-bioenergy supply chains: a review, Renewable Sustainable Energy Rev. 31 (2014) 657–670.

[23] A.D. Meyer, D. Cattrysse, J.V. Orshoven, Considering biomass growth and regeneration in the optimisation of biomass supply chains, Renewable Energy 87 (2016) 990–1002.

[24] N.Z. Atashbar, N. Labadie, C. Prins, Modeling and optimization of biomass supply chains: a review and a critical look, IFAC-PapersOnLine 49 (12) (2016) 604–615.

[25] L. Saikku, T.J. Mattila, Drivers of land use efficiency and trade embodied biomass use of Finland 2000–2010, Ecol. Ind. 77 (2017) 348–356.

[26] International Energy Agency, Energy Policies of IEA Countries – Finland. IEA. 2013.

[27] H. Gunnarsson, M. Rönnqvist, J.T. Lundgren, Supply chain modelling of forest fuel, Eur. J. Oper. Res. 158 (1) (2004) 103–123.

[28] D. Carlsson, M. Rönnqvist, Supply chain management in forestry—case studies at Södra Cell AB, Eur. J. Oper. Res. 163 (3) (2005) 589–616.

[29] S. D'Amours, M. Rönnqvist, A. Weintraub, Using operational research for supply chain planning in the forest products industry, INFOR: Inf. Syst. Oper. Res. 46 (4) (2008) 265–281.

[30] D. Carlsson, S. D'Amours, A. Martel, M. Rönnqvist, Supply chain planning models in the pulp and paper industry, INFOR: Inf. Syst. Oper. Res. 47 (3) (2009) 167–183.

[31] M. Rönnqvist, S. D'Amours, A. Weintraub, A. Jofré, E. Gunn, R.G. Haight, D. Martell, A.T. Murray, C. Romero, Operations research challenges in forestry: 33 open problems, Ann. Oper. Res. 232 (2015) 11–40.

[32] A. Marques, J. Rasinmäki, R. Soares, Amorim, Planning woody biomass supply in hot systems under variable chips energy content, Biomass Bioenergy 108 (2018) 265–277.

[33] S. Barreto, C. Ferreira, J. Paixao, B.S. Santos, Using clustering analysis in a capacitated location-routing problem, Eur. J. Oper. Res. 179 (3) (2007) 968–977.

[34] D. Bredström, M. Rönnqvist, Combined vehicle routing and scheduling with temporal precedence and synchronization constraints, Eur. J. Oper. Res. 191 (1) (2008) 19–31.

[35] M. Drexl, Synchronization in vehicle routing – a survey of VRPs with multiple synchronization constraints, Transp. Sci. 46 (3) (2012) 297–316.

[36] N. El Hachemi, I. El Hallaoui, M. Gendreau, L.M. Rousseau, Flow-based integer linear programs to solve the weekly log-truck scheduling problem, Ann. Oper. Res. 232 (1) (2015) 87–97.

[37] N. Matloff, "Introduction to discrete-event simulation and the SimPy language," Davis, CA, Dept of Computer Science, University of California at Davis, 2009.

[38] Gurobi Optimization Inc. (2015). Gurobi optimizer reference manual. [Online]. Available: http://www.gurobi.com.

Corresponding author
T.M. Pinho can be contacted at: tatianap@utad.pt