ANALYSING TRADE-OFFS IN CONTAINER LOADING: COMBINING LOAD PLAN CONSTRUCTION HEURISTICS WITH AGENT-BASED SIMULATION

Navonil Mustafee[1*]
Eberhard E. Bischoff[2]

[1]Centre for Innovation and Service Research
University of Exeter Business School
Building:One, Rennes Drive
Exeter EX4 4PU, UK

[2]College of Business, Economics and Law
Swansea University, Singleton Park
Swansea, SA2 8PP, Wales, UK

[*]Corresponding author contact details:
Email: n.mustafee@exeter.ac.uk;
Tel: +44 (0) 1392 725661
Fax: +44 (0) 1392 723210

ABSTRACT

In this paper we bring together two Operations Research techniques, Cutting and Packing Optimisation (CPO) and Simulation, and present a multi-methodology approach for analysing the trade-offs between loading efficiency and various important practical considerations in relation to the cargo - such as its stability, fragility, or possible cross-contamination between different types of items over time. The feasibility of this approach is demonstrated by considering a situation where the items to be loaded have differing degrees of perishability and where badly deteriorated items can affect those in their immediate vicinity (e.g. through the spread of mould). Our approach uses the output of the CPO algorithms to create agents that simulate the spread of mould through proximity-based interactions between the agents. The results show the trade-offs involved in container utilisation and the propagation of mould, without evidence of any correlation between them. The contribution of this research is the methodology and the feasibility study.

KEYWORDS: SIMULATION, OPTIMISATION, METHODOLOGY, TRANSPORT

1 INTRODUCTION

The efficient loading of cargo into freight containers - and, more generally, the proficient packing of smaller items into larger objects - has been a subject of intensive research for at least thirty years. George and Robinson (1980) were among the first to propose an algorithm for constructing a container loading plan. Their approach was heuristic in nature and based on the idea of building a series of 'walls' of items across the width and height of the container. Since then numerous different approaches have been developed for both the knapsack version of the problem - where the space available is fixed and loading all the cargo may not be possible - and, to a lesser extent, its bin-packing form - where all of the cargo involved must be stowed and a cost-effective way of using a set of containers is sought. In this paper the term Container Loading Algorithms (CLAs) is used to describe any approach which is designed to produce container loading plans.

CLAs are commonly used for a wide variety of container loading problems, for example, loading of cargo that consists of either identical items (completely homogenous cargo) or cargo consisting of a large number of different types of items relative to the total number of items (strongly heterogeneous cargo) or cargo comprising of a relatively few different types of items relative to the total number of items (weakly heterogeneous cargo), loading a consignment of goods into either a single container or multiple containers, etc. (Bischoff and Ratcliff, 1995). There are several studies that have proposed algorithm-based approaches aimed at homogeneous cargo (Han et al, 1989; George, 1992), strongly heterogeneous cargo (Gehring et al, 1990) and weakly heterogeneous cargo (George and Robinson, 1980; Morabito and Arenales, 1994; Ngoi et al, 1994).The majority of these studies have focused on a particular category of combinatorial optimisation problem - *the knapsack problem* – wherein the only parameters that are known/used are the dimensions of the cargo and the container and some measure of value associated with each item in the cargo (e.g., weight); given these inputs, the objective is to use CLAs to generate container loading plans that would allow the best possible utilisation of the container space (Bischoff and Ratcliff, 1995).

Much of the more recent work has moved away from pure knapsack or bin-packing formulations of the container loading problem and has paid increasing attention to various additional factors which may affect the task in practice. Orientation constraints on individual types of cargo and container weight capacity limits represent simple examples of such factors (Gehring and Bortfeldt, 2002). The literature has also used problem definitions which include the weight distribution within a container as a critical factor (Gehring and Bortfeldt, 1997; Davies and Bischoff, 1999; Eley, 2002) and aspects of cargo stability have been explicitly considered in several approaches as attributes of solution quality (Bortfeldt and Gehring, 1998; Terno et al, 2000; Bortfeldt and Gehring, 2001; Bortfeldt et al, 2003; Moura and Oliveira, 2005). Moreover, issues related to cargo fragility, in terms of constraints on the load bearing ability of items, have been taken into account by some authors (e.g. Bischoff, 2006).

Despite the considerable progress that has been made towards meeting the needs of practitioners, however, much work remains to be done before it can be claimed that the approaches suggested in the literature address adequately and fully the different problem scenarios which arise in practice. Combining load plan construction heuristics with simulation approaches is put forward here as a possible contribution towards this quest. More specifically, we focus on the CLAs (which are a particular category of Cutting and Packing Optimisation algorithms) and a specific simulation technique, namely, Agent-Based Simulation (ABS), and propose a multi-methodology approach for analysing trade-offs in relation to the loading of cargo into containers. Thus, while the CLAs determine the spatial co-ordinates of cargo placement inside a container, we intend to use the container loading plans as the input and simulate various cargo-related considerations. Two such considerations that are discussed in this paper are, (a) the stability of individual items of cargo as they are being transported (to simulate transportation of the container), and (b) cross-contamination between different items of cargo over time. Executing numerous experiments based on the proposed CLA-ABS approach, with each experiment using a different container loading plan, enables us to analyse the trade-offs between the aforementioned cargo-related considerations and loading efficiency.

The remainder of the paper is structured as follows. Section 2 presents an introduction to the agent-based modelling technique. This section also includes a discussion on related work (review of literature on the use of ABS for container management and container loading). Our multi-methodology approach combining CLA and ABS is discussed in Section 3. Sections 4 is devoted to the ABS model that we developed as part of our feasibility study; Section 5 discusses the experiments and presents some results. Section 6 is the concluding section of this paper. It highlights the contribution of the paper, discusses future work and draws the paper to a close.

2 AGENT-BASED SIMULATION

ABS is a simulation technique that models the overall behavior of a system through use of autonomous system components (also referred to as agents) which communicate with each other through use of messages. The behavior incorporated into an agent determines its role in the environment, its interaction with other agents, its response to message from other agents, and indeed whether its own behavior is adaptable. The agents usually have some properties and, unlike agent behavior, which may or may not be programmed to adapt itself, the properties usually change as a result of certain trigger points in the simulation - for example, time triggered changes, changes triggered by an agent's internal state change, changes triggered due to messages being received from other agent, etc. Since there are usually a number of self-governing agents in an environment, each with the aforementioned characteristics, the overall system state is determined by their dynamic interactions through time. In ABS, the simulation time increments in discrete time steps.

Figure 1 illustrates some fundamental concepts related to the execution of agents in an ABS environment (we recognise that the specific details will vary between different ABS implementations). An ABS can be thought of as a multi-threaded program (the threads are indicated by the six curved lines) that spawns from one main thread (indicated by the straight line). The ABS program being executed will generally have some global variables (indicated by the star that is associated with the main thread) and some variables that are specific to each thread (indicated by the six stars, each associated with one thread). Although the threads have read/write access to their local variables, they can only modify the global variables if they have acquired a "lock" on it (this lock is also referred to as the *semaphore*). The lock prevents multiple concurrent access to the shared global variable. We can intuitively think of every *agent instance* (*agent entity*) as a separate *thread of execution*; the local variables as *agent properties*; the global variables as *shared data*. Drawing parallels with the *Object-Oriented Programming (OOP)* terminology, each agent can be considered as a *Class* and each *agent instance* that is spawned from the main thread can be considered as an *Object of the Class*. We purposely draw reference to OOP since we consider that it is easier to program ABS if we think of agents as objects. Indeed, a modeler trying to implement a complex ABS without objects may find, at the end of the modeling exercise, that she/he has unintentionally re-invented some of the OOPs concepts (Shaliz, 2006). Figure 1 also depicts the trigger points (shown by the two light-colored circles) and the resultant inter-agent communication (depicted by the six dark circles). An ABS simulation advances time in discrete time steps, and therefore the underlying ABS software generally has an in-build time synchronisation/thread synchronisation logic that ensures concurrency of execution (indicted as "Time Sync" in the diagram), without causality errors.

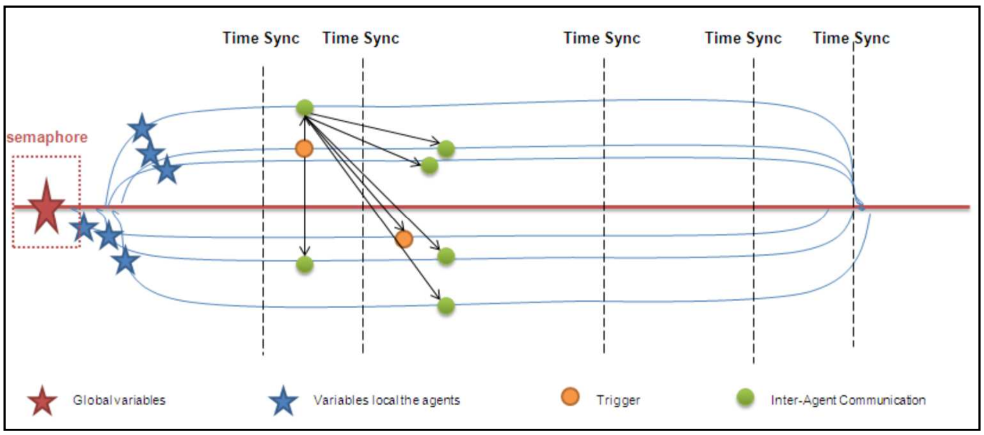


Figure 1: Execution of Agents in an ABS Environment

The description of agents and ABS presented above is based on our experience of working with this simulation technique. Macal and North have also discussed the characteristics of agents and ABS at length, and in the remainder of this paragraph we make reference to a couple of their papers. Macal and North (2006) consider agents to have six specific characteristics, namely, attributes, behavioral rules, memory, resources, decision making sophistication and rules to modify behavioral rules. The agent behavior, which is defined by simple rules, may be influenced by their interaction with other agents; and this "agent-by-agent and interaction-by-interaction" approach to modeling usually gives rise to emergent "patterns, structures and behaviors" that are not explicitly modeled in the system (Macal and North 2010).

ABS has several application areas. In healthcare, ABS has been used for modeling the spread of pathogens that are transmitted by direct contact (Hotchkiss et al, 2005); Stainsby et al (2009) have used agents to model hospital emergency departments; Sibbel and Urban (2001) have integrated agent-based approaches to classical simulation systems to enable better hospital management. In financial trading, ABS has been used to model New Electricity Trading Arrangements (NETA) in the UK (Bunn and Oliveira, 2001); application of this technique has been demonstrated in the financial market for price formation using realistic trade mechanism (Raberto et al, 2001). Agent-Based Social Simulation (ABSS) is the application of ABS in the social sciences context. Its use has been reported in the study of social dilemmas (e.g, Prisoner's Dilemma) by Gotts et al (2003); Downing et al (2001) report on the use of a prototype agent-based Integrated Assessment Model for understanding climate policy; ABSS has also been used for modeling of social interactions and influence (Marsella et al, 2004). Luo et al (2008) and Moulin et al (2003) have used ABS for crowd management. Applications of this technique have also been reported in supply chain management (Kaihara, 2003), in freight transportation (Davidsson et al, 2004), in the military (Cioppa et al, 2004), for emergency evacuations (Pan et al, 2007), in the management of the environment (Hare and Deadman, 2004), and in cell biology (Pogson et al, 2006), to cite just some further examples.

There have also been studies which use ABS in the context of container management. Henesey (2006), for instance, investigated the use of agent-based technologies to improve performance of container terminals. The use of an ABS architecture to solve the automatic container allocation problem in a port container terminal is described in Rebollo et al (2000). The use of agents to simulate and optimise cargo handling storage space in a maritime port is reported in Kefi et al (2009). ABS has been used to model the management of stakeholder relations in container terminals through use of agents that simulate different stakeholder behavior (Henesey et al, 2003). Henesey et al (2006) report on the use of the *SimPort* ABS tool to evaluate eight transshipment policies. Bin et al (2009) have used ABS, together with knowledge discovery, to model a container terminal logistics system through use of 14 kinds of agents. The project *Container World* has modeled both the business and the operational aspects of the container business in the UK though use of multi-agent methodology (Sinha-Ray et al, 2003).

Whilst the literature contains a number of ABS studies in the general area of port management, container terminal management and container business management, to the best of the authors' knowledge, there is no previous work on the use of ABS in the specific area of container loading. Thus, the authors are arguably among the first to report on the use of a multi-methodology approach that combines container load plan heuristics with ABS.

3 COMBINING CLA AND ABS: A MULTI-METHODOLOGY APPROACH

The purpose of the combined CLA and ABS approach to container loading is to identify the trade-off between loading efficiency and other practical considerations in relation to the cargo. In the research presented in this paper, we limit ourselves to the knapsack version of the problem, wherein the space available is fixed and loading all the cargo may not be possible. Also, the cargo consists of weakly heterogeneous items of rectangular shape – which will be referred to as 'boxes' for the sake of simplicity.

With the objective of better describing the application context pertaining to our proposed approach, we consider two problem scenarios– (a) the stability of the individual items of cargo as they are being transported *(cargo stability scenario)*, and (b) the possible cross-contamination between these items over time (*cargo cross-contamination scenario*). With respect to the latter scenario, the example considers a situation where the items to be loaded have differing degrees of perishability and where badly deteriorated items can affect those in their immediate vicinity through the spread of mould. We consider this example as a valid test case since, in the case of perishable goods, a supplier usually cares more about the decay of perishable goods than the average loading ratio and is willing to use more vehicles to prevent deterioration (Chen et al., 2009). For both scenarios we suggest that the individual items of cargo can be modeled as agents because they display three important characteristics consistent with most agents – (1)

they have properties, (2) they exhibit autonomous behavior, and (3) they interact with other agents which leads to more complex behavior (interaction is usually brought about by the spatiotemporal changes during transportation). Our argument for using ABS to model the aforementioned scenarios is summarised in Table 1. The table lists the three agent characteristics (column one) in relation to cargo stability (column two) and cargo cross-contamination (column three).

Table 1: Agent Characteristics for the Cargo Stability and the Cross-Contamination Scenarios

Agent Characteristics	Cargo Stability Scenario	Cargo Cross-Contamination Scenario (e.g., perishable items)
(1) Agents have properties	Dimension of the individual items of cargo (length, width, height).	Freshness index (i.e., the length of time after which the individual items of perishable goods will start developing mould). Dimension of the individual items of cargo (length, width, height).
(2) Agents exhibit autonomous behavior	The stability of an item is usually dependent on the dimensions of the item and other factors (e.g., weight).	The freshness index will usually decrease over time.
(3) Interaction between agents may result in more complex behavior	The stability of an individual item may also depend on the stability of the items that surround/support the item in question.	Perishable items may develop mould when they come into contact with other mould-affected items.

Having presented the rationale for using ABS to model the container cargo in our example scenarios, we now discuss our combined CLA-ABS approach. The methodology is applied in two phases and there are several iterations (refer to Figure 2 - left). In the first phase, CLAs are applied to generate a container loading plan and a corresponding container utilisation percentage. The load plan (refer to Figure 2 - right) should ideally be a text file that consists of the x, x' (x + length of box), y, y' (y + width of box), z and z' (z + height of box) co-ordinates of each item selected in the cargo (note that in our *knapsack version of the problem* not all items may be packed) and a corresponding freshness index. The container utilisation percentage is a function of the items selected by the CLAs. Thus, for the same mix of cargo to be loaded into a freight container, it is possible to have different container utilisation percentages based on the individual items selected by the different runs of the CLAs. The utilisation percentage will be an important input in the later stages of our methodology. The proposed methodology has no underlying dependency on any particular set of CLAs. The only requirement is that the CLAs being used in phase one provide output relating to the physical arrangement of the different items of cargo in the x, y and z axes (the container loading plan) and a corresponding container utilisation percentage. Optionally, the CLAs may also provide values that are related to other specific cargo-related considerations, for e.g., freshness index in our cargo cross-contamination scenario.

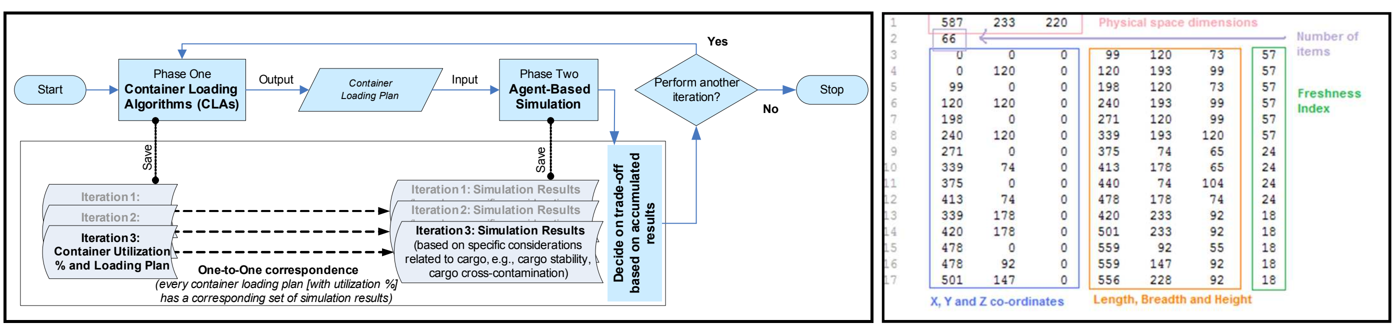


Figure 2: Flowchart of the Multi-Methodology Approach for Container Loading (left); Load plan generated by the CLAs (right)

In the second phase, the container loading plan generated in stage one can be used as an input to an ABS model (refer to Figure 2 - left). Our approach stipulates that the ABS model to be developed should have the following functionality: The ABS model needs to enable, (1) agents to be created for each item listed in the loading plan; (2) agents to be bestowed the mandatory dimension properties - length, width, and height (these can be calculated from the x, y, z axes values present in the container loading plan)– and one or more optional properties that may be present in the loading plan (such as the freshness index); (3) an agent's proximity-interactions with other agents to be determined based on the agents' x, y and z co-ordinates. Thus, the logic incorporated in the ABS model should be able to determine if, for example, two agents representing two items of cargo [box1, box2] are in contact with each other through a comparison on the x-axis (e.g., box1.x'==box2.x or box1.x==box2.x'), y axis and z axis. The mandatory dimension properties may also be used for 3D representation of the items of cargo (see section 4 for our implementation). Our multi-methodology approach has no underlying dependency on any specific ABS software.

The agent-based model can be executed to simulate the transport of cargo through time. The results to be collected will depend on the scenario being investigated. With regard to the two example problem scenarios being investigated in this study, result collection for cargo stability may involve the number of boxes that have fallen off the stack during transportation (i.e., upon completion of the simulation run); result collection for cargo cross-contamination scenario may involve the number of mould affected boxes. Collection of results completes phase two of our multi-methodology approach, and at this point we have a container utilisation percentage (output of CLAs) and a related set of results from the ABS. Again referring back to our example scenarios, a container utilisation of 89% may have resulted in 20 boxes having fallen off the stack during transportation and 45 boxes being affected by mould. The next paragraph describes the concept of phase one-phase two iterations in our methodology.

The purpose of our combined approach is to find the trade-off between container loading efficiency and other practical considerations in relation to the cargo. The considerations may be, (1) for the cargo stability scenario, to have an increased cargo steadiness (this is important for items containing fragile material), and (2) for the cargo cross-contamination scenario, to have the least number of mould affected boxes (this is important for perishable items). Since every execution of the CLAs in phase one will usually result in a different loading plan (and a corresponding utilisation percentage), and the resultant loading plan from phase one can be used for ABS of the container cargo in phase two, it follows that by conducting several of these phase one-phase two iterations we may have sufficient data to make a meaningful comparison and investigate the possible trade-offs. Using our cargo stability example, if a container utilisation of 67%, 78% and 90% in stage one results in 30, 18 and 25 boxes having fallen off their stack (the results of the ABS simulation), it may be worthwhile to use the 78% container utilisation loading plan rather than the one providing a 90% utilization. Similarly, if the application of our multi-methodology approach for the cross-contamination scenario presents us with ABS results of 14, 20 and 15 boxes being mould affected as a result of loading plans with 85%, 60% and 92% utilisation percentages, then the stakeholders are in a position to make an informed decision as to whether the container is to be loaded with the 85% or the 92% container utilisation loading plan. Finally, as with most simulations, the larger the number of phase one-phase two iterations (we do not use the term experiments in this connection since each iteration may involve several ABS experiments/trials when the model is stochastic), the greater the amount of data which is available to make an informed decision regarding the seletion of an appropriate loading plan.

4 THE SIMULATION MODEL

This section of the paper describes the ABS models that were used to experiment with phase two of our approach. The CLAs used in phase one are not presented here owing to page restrictions. However, these algorithms have been previously published and the reader is referred to Bischoff (2006). The scenarios modeled are the aforementioned ones - cargo stability and cargo cross-contamination. The cargo stability scenario presents an elementary proof-of-concept that uses an open-source physics engine to model the individual items of cargo (section 4.1). For the cross-contamination scenario, a 3-D ABS model has been developed using AnyLogic software (XJ Technologies, 2011) that simulates the propagation of mould in a freight container (section 4.2).

4.1 Example of cargo stability using a physics engine

In order to effectively simulate the stability of cargo during transportation, it may be necessary that the individual items be modeled such that they adhere to Newton's three laws of motion. The laws of motion deal with force and mass, and they provide the basis for understanding the effect that forces have (e.g., gravitational force, normal force, frictionless force) in determining the motion of an object (Cutnell and Johnson, 2001). In such a model the Newtonian motion that will be exhibited by the individual items of cargo will depend on its properties, e.g., length, width, height, mass/weight, contact friction, etc. The Newtonian motion can be modeled through use of a physics engine like Jinngine (Silcowitz-Hansen, 2010). Yet another requirement for modeling cargo stability is contact modeling for the detection of collision between objects. Some physics engines like Jinngine also provide collision detection support.

We argue that the collision detection among items of cargo which exhibit independent Newtonian motion will permit the simulation of cargo stability *only if* the individual items are modeled as agents in an ABS. This is because the agents can be simulated through time. Thus, a stack of boxes arranged inside a container (Figure 3 – left) can be simulated through time if, (a) every box is transformed into an agent – this may require the use of ABS software, (b) every box is bestowed the independent Newtonian motion (its motion will, in turn, will depend upon the agents' properties, e.g., dimensions, weight, etc.) – this may require the use of a physics engine, and (c) the proximity-based interactions among the boxes is captured using collision detection techniques – this can be accomplished through use of software that supports collision detection. The resultant simulation could potentially determine the cargo stability based on the number of boxes that may have fallen off the stack during transportation (when visually represented in 3D, this may look like Figure 3 –right).

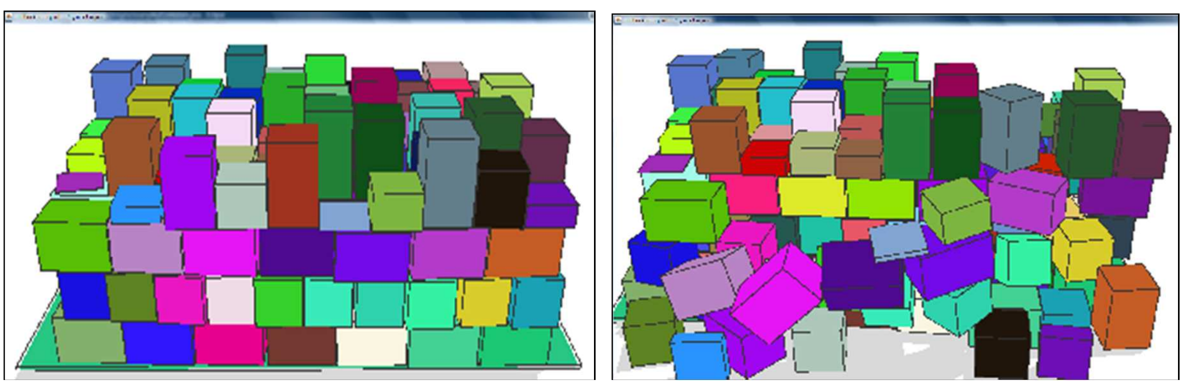


Figure 3: The stack of boxes while at rest (left) and subsequent to external stimulus being applied (right) – visualisations generated by extending Jinngine example programs (Silcowitz-Hansen, 2010)

It is to be noted here that the *proof-of-concept program* used to generate the screenshots above employs Jinngine for (b) and (c) – this program extends example programs that are available for download from the Jinngine website (Silcowitz-Hansen, 2010); it does not include any ABS agent functionality. As such we are unable to simulate the model through time. The screenshot presented in Figure 3 (right) is the result of an external stimulus (pull) being applied to the base of the container in an instant in time. Thus, the example is more akin to a 3D game that incorporates objects and collision detection. It is not a true ABS simulation.

Finally, although the boxes in our example program are not modeled as ABS agents, they do display the important characteristics that are found in most agents, namely, (a) they have properties (the boxes are of different dimensions and colors), (b) they display autonomous behavior (the boxes display independent Newtonian motion, which are, in turn, dependent on the properties mentioned in (a)), and (c) inter-agent interaction results in a more complex behavior (the collision among boxes will result in some boxes toppling over, which may, in turn, affect the other boxes inside the container). Future work could involve the integration of a piece of ABS software with Jinngine in order to perform rigid body simulation of items of cargo through time.

4.2 Modeling cargo cross-contamination using AnyLogic

The AnyLogic model developed by the authors is informed by the functionality proposed by the multi-methodology approach for phase-two ABS (refer to section 3). Namely, (1) the model reads the phase-one loading plan generated by the CLAs and creates a unique agent for each item of cargo listed in the loading plan; (2) each agent is provided with four properties – length, width, height and freshness index (the items of cargo/agents and the surrounding container is represented visually in 3-D space); (3) for every agent which is initialised, the ABS model identifies all the other agents surrounding the agent in question and establishes a relationship between them (this inter-agent relationship is used during simulation for inter-agent message passing). The model is run for 30 days in simulation time.

With regard to (3), in a conventional ABS model the agent interactions are usually defined by the modeler using package-specific functionalities. For example, AnyLogic, defines four pre-defined agent layouts in continuous space (random, ring, arranged and spring mass layouts) and two layouts in discrete space (random and arranged layouts) – these layouts are depicted in Figure 4 below (XJ Technologies, 2011). The agents are represented as dots or squares of different colors (red, black, blue) and the agent relationships are indicated by either lines (in case of agents in continuous space) or the physical placement of the agents (in case of agents in discrete space).

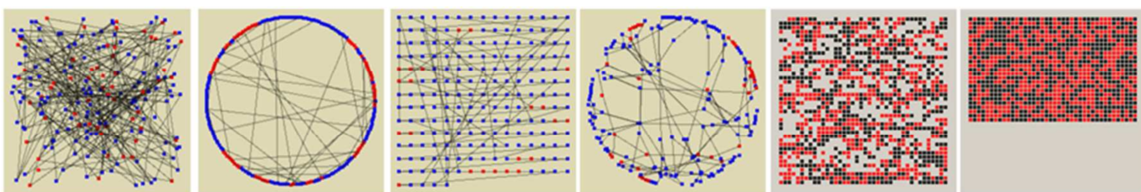


Figure 4: AnyLogic-defined arrangement of agents (from left to right): Random layout, Ring layout, Arranged layout, Spring Mass layout, Random layout (discrete space), Arranged layout (discrete space) (XJ Technologies, 2011).

However, this pre-defined agent arrangement and the pre-defined inter-agent relationship is inappropriate for our ABS model, since our agent arrangement will be defined by the container loading plan. Inter-agent relationships will then be created based on the physical proximity with the other agents. An example will make this obvious (refer to APPENDIX 1 for pseudo code). Figure 5 (left) shows a randomly-selected agent in question (red box), and at least eight other agents surrounding this agent; these proximity relationships are identified by the model based on the x, x', y, y', z and z' coordinates of the different items of cargo, and these inter-agent relationships are subsequently stored in the model.

Every agent has an initial freshness index generated in phase one. This freshness index is a randomly generated integer value between 5 and 60 and it is assigned to different Box Types. For example, if 10 boxes of a particular Box Type (with a freshness index of 14) have been selected by the CLAs, each of these will have a freshness index of 14. The index is used to denote the number of simulated days for which the contents of the box will remain mould free. This is not necessarily related with the units' expiration date. Thus, as the simulation progresses in time, at the end of every simulated day, the freshness value associated with every agent is decreased by one (as long as it is positive). When the freshness level of a particular box reaches zero, it is said to be mould affected. We have used the notion of a freshness index in our simulation based on evidence in the literature of contamination and/or decay of perishable products subsequent to the expiration of well-defined time units; also, the time units may differ from one product to the other. For example, the mould infection (fungal spoilage) for fruits in transit for up to 14 days can be as high as 100% with the average mould rate depending on the fruit being transferred

(Tournas and Katsoudas, 2005). Decay fungi are easily spread throughout the shipping container (Ashby, 1995) and at different rates, for example, berries in a shipping container have an average mould rate of 80% in 14 days of transit, citrus fruits have an average of 45%, etc. (Tournas and Katsoudas, 2005). In the 3-D visualisation the color of the box is changed to green so as to enable easier identification (Figure 5 – right). The transition of the agents is modelled by a state chart (refer to Appendix 2).

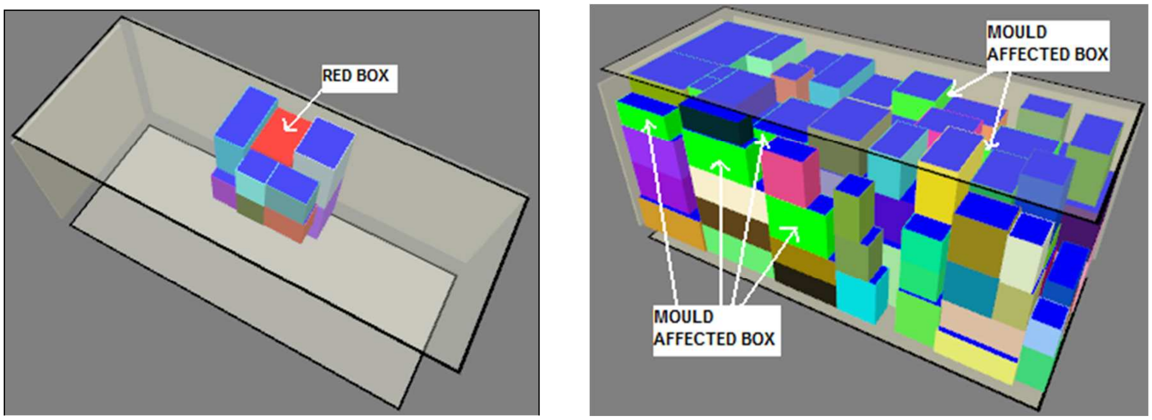


Figure 5: Detection of proximity among agents (left); Mould affected boxes during simulation (right)

When a box becomes mould affected it starts to cross-contaminate other boxes that physically surround it. This cross-contamination is reflected in the model by the proximity-based interactions sent by the agent (representing the mould affected box) to all the surrounding agents. Thus, in Figure 5 (left), if the red box/agent were to become mould affected, it will immediately send messages to all the eight boxes/agents surrounding it. The message is a signal to the other agents (each of whom have their own copy of a freshness index that decreases over time) that they have now been contaminated. Upon receiving this message, each agent immediately decreases the existing value of its freshness index by 1 (if freshness index > 0). This is in addition to the drop of 1 unit of freshness that is applicable at the end of each simulated day. The logic of the model, as described above, is repeated till the end of the simulated period. After the successful execution of the ABS several pieces of data are collected, including, the number of mould affected boxes, the maximum freshness value associated with an item of cargo, the average cargo freshness, a histogram associated with freshness data. This completes one ABS phase-two iteration. As has been described earlier, our multi-methodology approach involves several such iterations being performed, each with a different container loading plan, so as to enable the stakeholder to decide on a trade-off between container efficiency and considerations such as cargo stability, etc. Experiments and results are described next.

5 EXPERIMENTS AND RESULTS

This section describes the application of the proposed two-stage multi-methodology agent-based approach to the cargo cross-contamination scenario. Phase one of our methodology uses the CLAs that were originally proposed by Bischoff (2006) for the *3/B/O/R category of cutting and packing problems* (Dyckhoff, 1990). These CLAs are henceforth referred to as the *benchmark scenario generation algorithms*. The 3/B/O/R category describes the problem space in which a weakly heterogeneous mix of goods is to be loaded into a single container, wherein both the goods and the container are rectangular in shape and have known dimensions (Bischoff, 2006). The heterogeneous cargo is modelled as individual "boxes" that belong to different "box types". In (Bischoff, 2006) the factors that distinguish the various box types are the boxes' physical dimensions, i.e., length, width, and height, their weight and their load bearing strength. For the purpose of the cargo cross-contamination scenario presented in this paper, weight and load bearing strength are ignored and a new property – *freshness index*– is introduced. Given a weakly heterogeneous mix of box types as the input, the *benchmark 3/B/O/R scenario generation* *algorithms* produces hundreds of alternative container loading plans, each having a corresponding container utilisation rate. Given a specific mix of box types (subsequently referred to as *"Box Type Mix", or BTM for short*), using the benchmark algorithms we are able to determine the most efficient loading pattern by comparing the container utilisation rates associated with the various loading patterns that are output by the algorithm.

For our cross-contamination scenario experiments we have used a total of four BTMs, namely, BTM**20**, BTM**40**, BTM**70** and BTM**100***,* each of which represents a specific *Box Type Mix* with 20, 40, 70 and 100 box types respectively. Using the *benchmark 3/B/O/R scenario generation algorithms* we generated 100 container loading plans for each of our four BTMs. We then ranked the container utilisation rates for the aforementioned loading plans, and selected three specific loading plans for every BTM. These are as follows (where, ***x***=20,40,70,100): (a) loading plan with the lowest container utilisation (BTM***x****min*); (b) loading plan with utilisation percentile rank of 50 (BTM***x****mid*); and (c) loading plan with the highest container utilisation (BTM***x****max*). Each BTM is a unique experiment group; each group comprises of three separate iterations, wherein each phase-two iteration uses a unique container loading plan (BTM***x****min*, BTM***x****mid* or BTM***x****max*). The iterations will henceforth be identified by the BTM number and a corresponding loading plan. For example, BTM**40***max* will refer to the iteration using the container loading plan with the highest container utilisation (*max*) and which is associated with the experiment group BTM**40**.

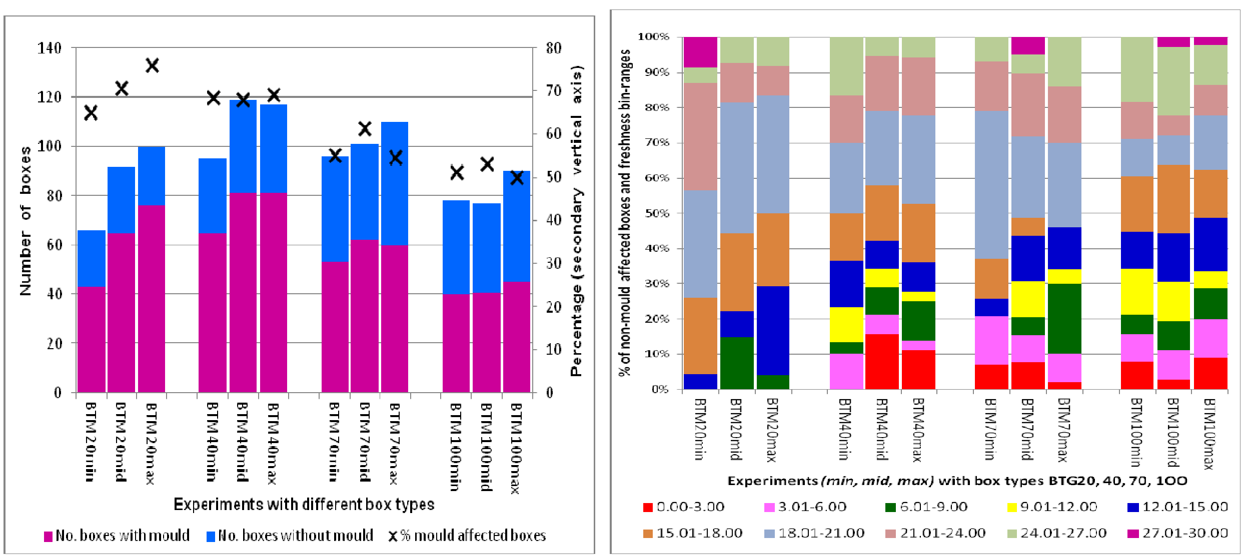


Figure 6: % mould affected boxes, grouped by BTM (left); Histogram showing freshness % of boxes not affected by mould, grouped by BTM (right)

Figure 6 (left) illustrates, for each of the three iterations pertaining to the four BTM experiments, the number and the corresponding percentage of boxes affected with mould. Since the agents represent boxes that are in physical contact with each other (and therefore susceptible to cross-contamination), it is therefore the case that, following the logic of the program, the majority of the boxes will be contaminated before the expiry of their freshness index. Irrespective of the box layout pattern used, 60-80% of the boxes will become mould affected by the end of the 30-day simulated run. It is important to note that although the results for all the BTMs are presented in the same figure (Figure 6 – left and right), comparison cannot be made across BTMs since the freshness index of the items of cargo (generated using random numbers which are assigned to box types) changes from one experiment type to the next.

Our experiments show clearly the existence of trade-offs between container loading efficiency and the proportion of boxes affected with mould. For example in Figure 6 (left), in relation to the BTM**70** experiments, the three phase-one loading plans selected had utilisation efficiencies of 72.10% [Total boxes selected = 96] (BTM**70***min*), 81.68% [101 selected] (BTM**70***mid*) and 86.70% [110 selected] (BTM7**0***max*), respectively; the corresponding percentages of mould affected boxes produced by the phase-two simulation were 55.21%, 61.39% and 54.55% (refer to the secondary vertical axis shown in Figure 6 – left). Intuitively, it might seem that a layout pattern with a greater space utilisation (e.g., BTM7**0***max*) would result in a higher number of mould-affected boxes (when, for example, compared to BTM**70***min*), simply because the boxes are more cramped together. However, there are several other factors which determine the percentage of boxes affected. One such factor is the number of boxes used in the layout pattern. To take an extreme example, we can have a case where only 5 very large boxes have been selected by the CLA to achieve 95% space utilization; in contrast, in another layout, the CLA identifies 500 small boxes and achieves 87% space utilization. Since every box, when its freshness index reaches zero, will begin to contaminate the surrounding boxes, the spread of mould in the latter layout (with the lower space utilization) will arguably be more amplified due to the fact that in this arrangement there are many boxes which are in contact with each other. Another factor can be the initial freshness index associated with boxes that are selected by the CLA algorithm. As has been mentioned, the freshness index is a randomly generated value between 5 and 60 and all boxes of the same type have the same freshness index. Thus, the propagation of mould also depends directly on the freshness attribute of the specific boxes being selected by the CLA algorithm – and this may change from one layout plan to the other.

Our multi-methodology approach has the potential to help the stakeholder to take an informed decision on the loading plan to use, as it provides both the container utilisation percentage (from stage one) and the corresponding simulation data (from stage two) from multiple iterations. Let us consider as a first example the BTM**70** experiments. The stakeholder, having seen the results of three iterations, is likely to choose the loading plan that offers 86.70% utilisation efficiency (BTM**70***max*) in preference to the other two – which provide lower space utilisation in conjunction with a higher proportion of spoilt boxes. It should be noted that the results of the experiments show no evidence of either positive or negative correlation between container utilisation percentage and the percentage of boxes affected with mould. For example, in the case of BTM**40**, the percentage of mould affected boxes would be roughly the same for the BTM**40***min* (68.42%), BTM4**0***mid* (68.07%) and BTM**40***max* (69.23%) loading plans.

Figure 6 (right) presents a histogram of freshness values for each of the three iterations pertaining to the four BTM experiments. It shows that for all the three loading plans that were simulated for BTM**20**, the items that had not already developed mould, would remain mould free for at least 6 days (note the freshness/bin range for BTM**20***min* starts at bin range 12.01-15.00; BTM**20***mid* and BTM**20***max* start at bin range 6.01-9.00). It further shows that if the BTM**20***min* loading plan is selected, then approximately 95.65% of items will remain relatively fresh (bin/freshness range: 15.00 to 30.00) by the end of the 30-day simulation (logical time) – and this value is at least 18% higher when compared to the corresponding freshness values of the alternate loading plans (BTM**20***mid*=77.78%; BTM**20***max*=70.83%). Here, therefore, the decision on which loading plan to choose involves a trade-off between loading efficiency and the risk of cargo deterioration and the stakeholder is likely to make it on the basis of practical business considerations.

6 CONCLUSION

In this paper we bring together two Operations Research techniques - *Cutting and Packing Optimisation (CPO)* and *Simulation* - and apply them in the context of container loading. More specifically, we focus on a particular category of CPO algorithms - *Container Loading Algorithms (CLAs)* - and a specific simulation technique - *Agent-Based Simulation (ABS)* - and propose a multi-methodology approach in relation to the loading of cargo into containers. The application of our approach will enable the stakeholders to analyse the trade-offs between loading efficiency and other cargo-related considerations. A feasibility study is conducted in order to test the approach. It investigates the trade-off between container utilisation and the cross-contamination among boxes containing perishable goods. We consider this example as a valid test case since, in the case of perishable goods, a supplier usually cares more about the decay of perishable goods than the average loading ratio and is willing to use more vehicles to prevent deterioration (Chen et al., 2009).

The multi-methodology approach is realised through the use of the *benchmark 3/B/O/R scenario generation algorithms* (Bischoff, 2006) and the development of an ABS model in AnyLogic (XJ Technologies, 2011). The results of the experiments demonstrate the trade-offs involved between container utilisation and the number of mould affected boxes. The results also show that there is no correlation between the aforementioned variables. The primary contribution of this research is the statement of the multi-methodology agent-based approach to container loading (section 3) and the feasibility study on cargo efficiency and cross-contamination (sections 4.2 and 6). This work has also given pointers for future research into simulation of container stability and has presented an elementary proof-of-concept that uses an open-source physics engine to model cargo stability of individual items of cargo (section 4.1).

ACKNOWLEDGEMENT

An earlier version of this paper was presented at the 2011 Winter Simulation Conference, Phoenix, Arizona (Mustafee and Bischoff, 2011).

REFERENCES

Ashby BH (1995). Protecting perishable foods during transport by truck. Handbook no. 669, US Department of Agriculture, Agricultural Marketing Service. *Available online http://www.ams.usda.gov/AMSv1.0/getfile?dDocName=STELDEV3021003* [last accessed 29 June 2012].

Bin L, Wen-Feng L and Yu Z (2009). Study on Modeling of Container Terminal Logistics System Using Agent-Based Computing and Knowledge Discovery. *International Journal of Distributed Sensor Networks* **5(1):**36-36.

Bischoff E E (2006). 3-D Packing of Items with Limited Load Bearing Strength. *European Journal of Operational Research* **168(3):**952-966.

Bischoff E E and Ratcliff M(1995). Issues in the Development of Approaches to Container Loading. *Omega* **23(4):**377-390.

Bortfeldt A and Gehring H (1998). Ein Tabu Search-Verfahren für Containerbelade-probleme mit schwach heterogenem Kistenvorrat. *Operations Research Spektrum* **20:**237–250.

Bortfeldt A and Gehring H (2001). A hybrid genetic algorithm for the container loading problem. *European Journal of Operational Research* **131(1):**143–161.

Bortfeldt A, Gehring H and Mack D (2003). A parallel tabu search algorithm for solving the container loading problem. *Parallel Computing* **29(5):**641–662.

Bunn D W and Oliveira F S (2001). Agent-based simulation-an application to the new electricity trading. *IEEE Transactions on Evolutionary Computation* **5(5):**493-503.

Chen H-K, Hsueh C-F and Chang M-S. (2009). Production scheduling and vehicle routing with time windows for perishable food products. *Computers & Operations Research* **36(7)**: 2311-2319.

Cioppa T M, Lucas T W and Sanchez S M (2004). Military applications of agent-based simulations. *In Proceedings of the 2004 Winter Simulation Conference*, pp. 171-180.

Cutnell J D and Johnson K W (2001). Forces and Newton's laws of motion. *Physics*. John Wiley & Sons: New York.

Davidsson P, Henesey L, Ramstedt L, Törnquist J and Wernstedt F (2004). Agent-Based Approaches to Transport Logistics. *In Proceedings of the 2004 Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004) – Workshop on Agents in Traffic and Transportation*.

Davies A P and Bischoff E E (1999). Weight distribution considerations in container loading. *European Journal of Operational Research* **114(3):**509–527.

Downing T E, Moss S and Pahl-Wostl C (2001). Understanding Climate Policy Using Participatory Agent-Based Social Simulation. *Lecture Notes in Computer Science* **1979/2001:**127-140.

Dyckhoff H (1990). A typology of cutting and packing problems. *European Journal of Operational Research* **44(2):**145–159.

Eley M (2002). Solving container loading problems by block arrangement. *European Journal of Operational Research* **141(2):**393–409.

Gehring H and Bortfeldt A (1997). A genetic algorithm for solving the container loading problem. *International Transactions in Operational Research* **4(5/6):**401–418.

Gehring H and Bortfeldt A (2002). A parallel genetic algorithm for solving the container loading problem. *International Transactions in Operational Research* **9(4):**497–511.

Gehring H, Menschner K and Meyer M (1990). A computer-based heuristic for packing pooled shipment containers. *European Journal of Operational Research* **44(2):**277–288.

George J A (1992). A method for solving container packing for a single size box. *Journal of the Operational Research Society* **43(4):**307–312.

George J A and Robinson D F (1980). A heuristic for packing boxes into a container. *Computers & Operations Research* **7(3):**147–156.

Gotts N M, Polhill J G and Law A N R (2003). LawAgent-Based Simulation in the Study of Social Dilemmas. *Artificial Intelligence Review* **19(1):**3-92.

Han C P, Knott K and Egbelu P J (1989). A heuristic approach to the three-dimensional cargo-loading problem. *International Journal of Production Research* **27(5):**757–774.

Hare M and Deadman P (2004). Further towards a taxonomy of agent-based simulation models in environmental management. *Mathematics and Computers in Simulation* **64(1):**25-40.

Henesey L E (2006). *Multi-Agent Systems for Container Terminal Management*. PhD thesis, Blekinge Institute of Technology. Blekinge Institute of Technology Doctoral Dissertation Series No 2006:08.

Henesey L E, Davidsson P and Persson J A (2006). Agent Based Simulation Architecture for Evaluating Operational Policies in Transshipping Containers. *Lecture Notes in Computer Science* **4196/2006**:73-85.

Henesey L E, Notteboom T E and Davidsson P (2003). Agent-based simulation of stakeholders relations: An approach to sustainable port terminal management. *In Proceedings of the 2003 International Association of Maritime Economists Annual Conference*, pp. 314-331.

Hotchkiss J R, Strike D G, Simonson D A, Broccard A F and Crooke P S (2005). An agent-based and spatially explicit model of pathogen dissemination in the intensive care unit. *Critical Care Medicine* **33(1):**168-176.

Kaihara T (2003). Multi-agent based supply chain modelling with dynamic environment. *International Journal of Production Economics* **85(2):**263-269.

Kefi M, Ghedira K, Korbaa O and Yim P (2009). Container handling using multi-agent architecture. *International Journal of Intelligent Information and Database System* **3(3):**338-360.

Luo L, Zhou S, Cai W, Low M Y H, Tian F, Wang Y, Xiao X and Chen D (2008). Agent-based human behavior modeling for crowd simulation. *Computer Animation and Virtual Worlds* **19(3-4):**271–281.

Macal C M and North M J (2006). Tutorial on Agent-Based Modeling and Simulation Part 2: How to Model with Agents. *In Proceedings of the 2006 Winter Simulation Conference*, pp. 73–83.

Macal C M and North M J (2010). Tutorial on agent-based modelling and simulation. *Journal of Simulation* **4(3):**151–162.

Marsella S C, Pynadath D V and Read S J (2004). PsychSim: Agent-based modeling of social interactions and influence. *In Proceedings of the International Conference on Cognitive Modeling, Pittsburgh*, pp. 243-248.

Morabito R and Arenales M (1994). An and/or-graph approach to the container loading problem. *International Transactions in Operational Research* **1(1):**59–73.

Moulin B, Chaker W, Perron J, Pelletier P, Hogan J and Gbei E (2003). MAGS Project: Multi-agent GeoSimulation and Crowd Simulation. *Lecture Notes in Computer Science* **2825/2003:**151-168.

Moura A and Oliveira J F (2005). A grasp approach to the container loading problem. *IEEE Intelligent Systems* **20(4):**50–57.

Mustafee, N. and Bischoff, E. E. (2011). "A Multi-methodology Agent-based Approach for Container Loading". In *Proceedings of the 2011 Winter Simulation Conference (WSC 2011),* Phoenix, Arizona, Dec 11-14, 2011. pp. 234-245.

Ngoi B K A, Tay M L and Chua E S (1994). Applying spatial representation techniques to the container packing problem. *International Journal of Production Research* **32(1):**111–123.

Pan X, Han C S, Dauber K and Law K H (2007). A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations. *AI & Society* **22(2):**113-132.

Pogson M, Smallwood R, Qwarnstrom E and Holcombe M (2006). Formal agent-based modelling of intracellular chemical interactions. *Biosystems* **85(1):**37-45.

Raberto M, Cincottia S, Focardib S M and Marchesic M (2001). Agent-based simulation of a financial market. *Physica A: Statistical Mechanics and its Applications* **299(1-2):** 319-327.

Rebollo M, Julian V, Carrascosa C and Botti V (2000). A Multi-Agent System for the Automation of a Port Container Terminal. *In Proceeding of the Autonomous Agents 2000 Workshop on Agents in Industry, Barcelona*.

Shaliz C R (2006). Methods and techniques of complex systems science: An overview. In: Deisboeck T S and Kresh J Y (eds). *Complex Systems Science in Biomedicine.* Springer: New York.

Sibbel R and Urban C (2001). Agent-based modeling and simulation for hospital management. In: Saam N J and Schmidt B (eds). *Cooperative Agent: Applications in the Social Sciences*. Kluwer Academic Publishers: The Netherlands.

Silcowitz-Hansen M (2010). Jinngine, a Physics Engine Written In Java. *http://code.google.com/p/jinngine/*, accessed 12 November 2012.

Sinha-Ray P, Carter J, Field T, Marshall J, Polak J, Schumacher K, Song D, Woods J and Zhang J (2003). Container World: Global agent-based modelling of the container transport business. *In Proceedings of the 4th Workshop on Agent-Based Simulation*.

Stainsby H, Taboada M and Luque E (2009). Towards an Agent-Based Simulation of Hospital Emergency Departments. *In Proceedings of the 2009 IEEE International Conference on Services Computing*, pp. 536 - 539.

Terno J, Scheithauer G, Sommerweiss U and Riehme J (2000). An efficient approach for the multi-pallet loading problem. *European Journal of Operational Research* **123(2):**372–381.

Tournas VH and Katsoudas E. (2005). Mould and yeast flora in fresh berries, grapes and citrus fruits. *International Journal of Food Microbiology*, **105(1)**:11-17.

XJ Technologies. 2011. XJ Technologies Homepage – AnyLogic. Accessed January 6, 2011. *http://www.xjtek.com/*, accessed 12 November 2012.

APPENDIX 1:

Pseudo Code for establishing inter-agent relationship between agents (note the code has been simplified)

```
//Read input file (Figure 4) generated by the CLA

//Create container space using length, breath and height dimensions
Read line 1 from input file

//Read the number of agents to be created
totAgents = [value indicated in line 2 of the input file];
numAgent = 0;

//Initialise boxes based on CLA-generated values (line 3 onwards); boxes are added to agent collection
//Note: In the code below "Boxes" is a physical manifestation of the agent – It has all the agent
characteristics and also dimensions. The dimensions are required for determining proximity between agents
and also for 3-D visualisation.
Do
{
  numAgent++;
  Boxes box = me.add_boxObj(xPos, yPos, zPos, length, breadth, height,
  random.nextInt(256), random.nextInt(256), random.nextInt(256), getFreshness,
  isItemPerishable);
}

While (totAgents> numAgent)

//Establish inter-agent relationship by looping through the agent collection

//loop through each agent
for(Agent agent : me.environment.getAgentCollection())
{
 //picking up one agent for comparing with all the other agents in the collection
 Boxes tempAgent = (Boxes)agent;

  //loop through all agents in the collection
  for(Agent agentTest : me.environment.getAgentCollection())
    {
      //picking up one agent from the collection for comparison with tempAgent
      Boxes tempAgentTest = (Boxes)agentTest;

      //if tempAgent is not the same as tempAgentTest
      if(tempAgent != tempAgentTest)
      {
         //check if tempAgentTest & tempAgent co-ordinates match at any point in x axis
         if((tempAgent.xPos+tempAgent.length==tempAgentTest.xPos) ||
     (tempAgent.xPos==tempAgentTest.xPos+tempAgentTest.length))
     {
       //check if co-ordinates match at any point in y axis
            if(((tempAgent.yPos >= tempAgentTest.yPos) &&
            (tempAgent.yPos <= tempAgentTest.yPos+tempAgentTest.breadth)) ||
            ((tempAgent.yPos+tempAgent.breadth>=  tempAgentTest.yPos)  &&  (tempAgent.yPos+tempAgent.breadth
            <= tempAgentTest.yPos+tempAgentTest.breadth)) || ((tempAgentTest.yPos>= tempAgent.yPos) &&
            (tempAgentTest.yPos <= tempAgent.yPos+tempAgent.breadth)))
        {
           //check if co-ordinates match at any point in z axis
               if(((tempAgent.zPos >= tempAgentTest.zPos) &&
               (tempAgent.zPos <= tempAgentTest.zPos+tempAgentTest.height)) ||
               ((tempAgent.zPos+tempAgent.height>= tempAgentTest.zPos) &&
               (tempAgent.zPos+tempAgent.height <= tempAgentTest.zPos+tempAgentTest.height)) ||
               ((tempAgentTest.zPos>= tempAgent.zPos) &&
               (tempAgentTest.zPos <= tempAgent.zPos+tempAgent.height)))
           {
          //THE TWO AGENTS ARE PHYSICALLY PROXIMATE (INTER-AGENT RELATIONSHIP HAS BEEN
                  ESTABLISHED
                 tempAgent.connectTo(tempAgentTest);
```

```
              //do not execute the remaining statements in the loop
              continue;

           } //end if statement - z axis
          } //end if statement – y axis
         } //end if statement – x axis


         //similarly, check if co-ordinates match at any point in z axis
         If ()
         {
           //check if co-ordinates match at any point in x axis
           If ()
           {
              //check if co-ordinates match at any point in y axis
              If ()
              {
                 //THE TWO AGENTS ARE PHYSICALLY PROXIMATE (INTER-AGENT RELATIONSHIP HAS BEEN
                 ESTABLISHED
                 tempAgent.connectTo(tempAgentTest);

                 //do not continue with the rest of the statements
                 continue;

            } //end if statement - y axis
          } //end if statement – x axis
         } //end if statement – z axis


         //similarly, check if co-ordinates match at any point in y axis
         If ()
         {
            //check if co-ordinates match at any point in x axis
            If ()
            {
                //check if co-ordinates match at any point in z axis
                If ()
                {
                   //THE TWO AGENTS ARE PHYSICALLY PROXIMATE (INTER-AGENT RELATIONSHIP HAS BEEN
                   ESTABLISHED
                   tempAgent.connectTo(tempAgentTest);

                   //do not continue with the rest of the statements
                   continue;

            } //end if statement - z axis
           } //end if statement – x axis
          } //end if statement – y axis
      } // end if statement – comparing tempAgent with tempAgentTest
   } // end of inner for loop
 } // end of outer for loop
```

APPENDIX 2:

Agent State Chart and Pseudo Code for inter-agent message transfer

State Chart

The transition of the agents is modelled by the state chart below (Figure 7). At the start of the simulation the state of every agent is *"StateFresh"*. Following the modelling assumption stated in the paper (in the model we have assumed that when the freshness level of a particular box reaches zero, it is mould affected), when the freshness value of an agent becomes zero, the agent state changes from *"StateFresh"* to *"StateMould"* and it sends messages (function: *sendMessagetoAgents*) to the agents with which it is in physical contact to indicate its change of state. The message is a signal to the other agents (each of whom has its own copy of freshness index that decreases over time) that they have now been contaminated. Upon receiving this message, each agent immediately decreases its existing value of freshness index by 1 (if *freshness index > 0*). This is in addition to the drop of 1 unit of freshness that is applied to every agent at the end of each simulated day.

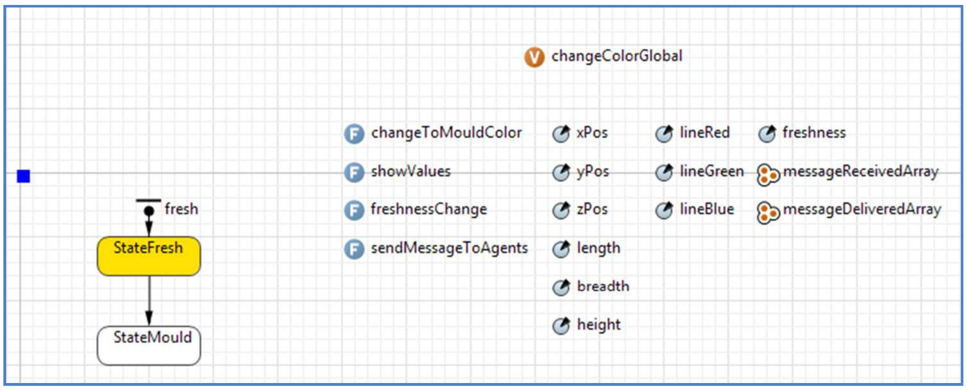


Figure 7: State chart

Pseudo Code

```
//Note: In the code below "Boxes" is a physical manifestation of the agent - It has all the agent characteristics and also dimensions. The dimensions are required for determining proximity between agents and also for 3-D visualisation.
//Note: At the start of the simulation all the agents are in StateFresh but they have different FreshnessIndex.

Do while (current_simulation_time < simulation_end_time)
{
    //loop through each agent
    for(Agent agent : me.environment.getAgentCollection())
    {
        //picking up one agent
        Boxes tempAgent = (Boxes)agent;

        //check the agent state
        //If agent is presently in StateFresh then decrease freshness by one; If agent is in
          StateMould then do take any action and select another agent.
          if (tempAgent.state == StateFresh)
          {
            //reduce the value of the variable freshness by one (this is achieve by the function
            freshnessChange() - refer to the State Chart above).
            tempAgent.freshnessChange();

            //check to see if the agent is now in StateMould  (since freshness has now reduced by 1)
            if(tempAgent.freshness = 0)
            {
                //The agent is mould affected.
```

```
                //Change the current state of the agent to StateMould
                tempAgent.state = StateMould;

                //Change the color of the mould-affected box to green (this is achieved by calling the
                function changeToMouldColor() – refer to the State Chart above).
                tempAgent.changeToMouldColor();

                //Send message to all the agents that are connected to this particular agent and inform
                them of the change in agent state (this is achieved by calling the function
                sendMessageToAgents() – refer to the State Chart above).
                tempAgent.sendMessageToAgents();

             } // end of if
          }
          else
          {
            //Agent is in StateMould – no action required. Select the next agent in the Agent collection.
            continue;
          }
      } // end of loop related to agent collection

    //increment the current simulation time by 1
    current_simulation_time= current_simulation_time+1;

} // end of Do While
```

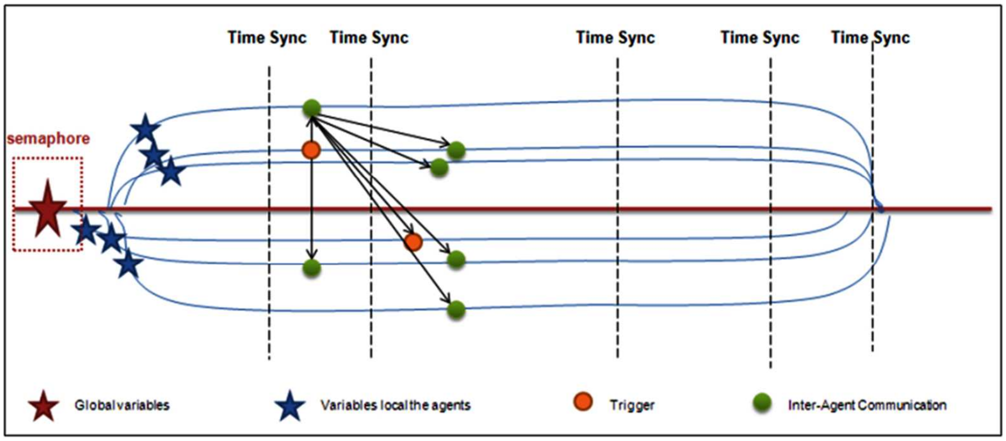


Figure 1: Execution of Agents in an ABS Environment

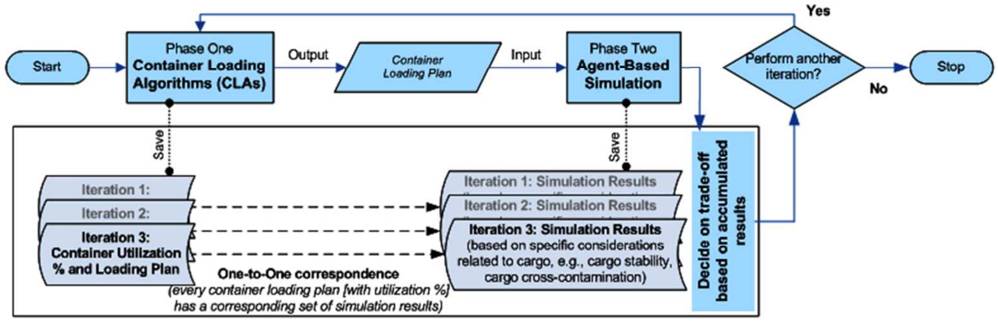
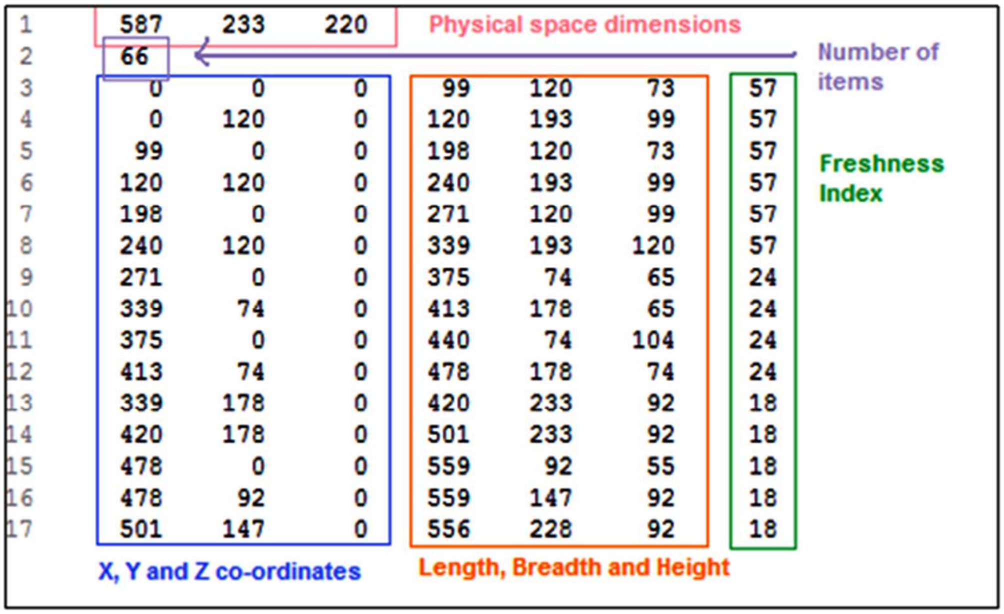


Figure 2: Flowchart of the Multi-Methodology Approach for Container Loading (left)



587	233	220				
66						
0	0	0	99	120	73	57
0	120	0	120	193	99	57
99	0	0	198	120	73	57
120	120	0	240	193	99	57
198	0	0	271	120	99	57
240	120	0	339	193	120	57
271	0	0	375	74	65	24
339	74	0	413	178	65	24
375	0	0	440	74	104	24
413	74	0	478	178	74	24
339	178	0	420	233	92	18
420	178	0	501	233	92	18
478	0	0	559	92	55	18
478	92	0	559	147	92	18
501	147	0	556	228	92	18

Figure 2: Load plan generated by the CLAs (right)

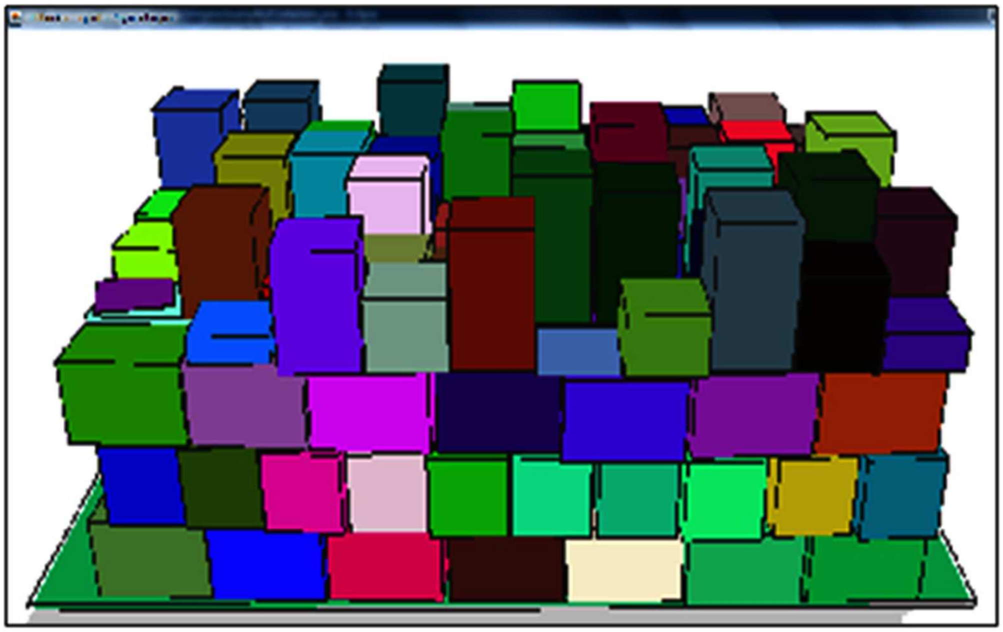


Figure 3: The stack of boxes while at rest (left)

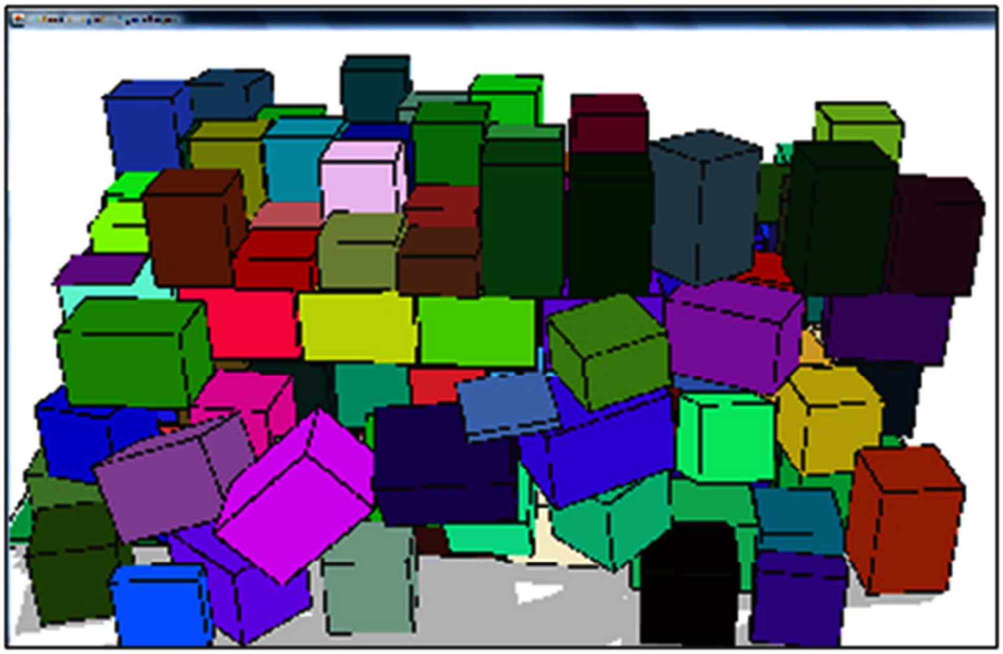


Figure 3: subsequent to external stimulus being applied (right) – visualisations generated by extending Jinngine example programs (Silcowitz-Hansen, 2010)

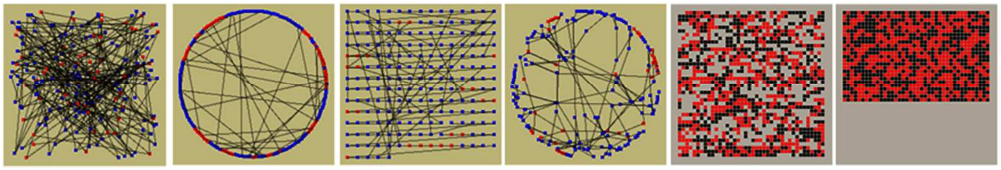


Figure 4: AnyLogic-defined arrangement of agents (from left to right): Random layout, Ring layout, Arranged layout, Spring Mass layout, Random layout (discrete space), Arranged layout (discrete space) (XJ Technologies, 2011).

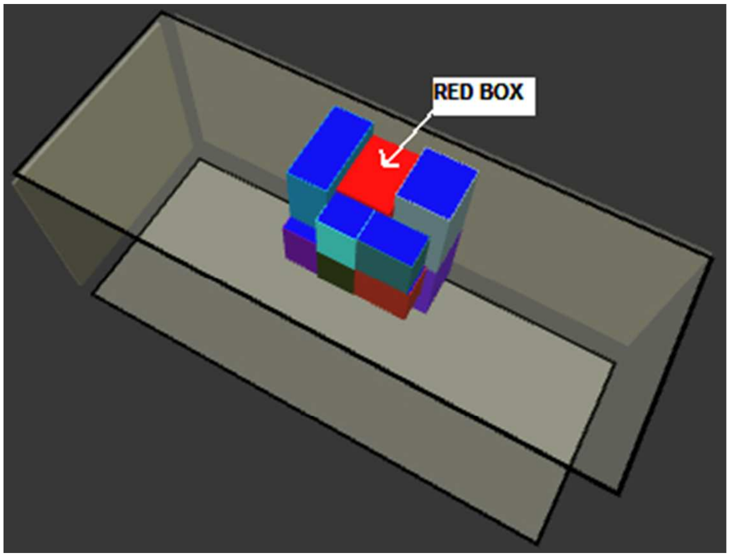


Figure 5: Detection of proximity among agents (left)

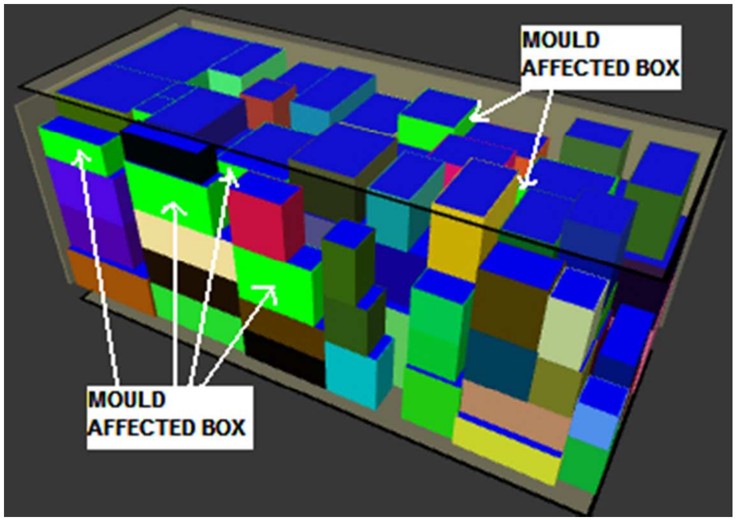


Figure 5; Mould affected boxes during simulation (right)

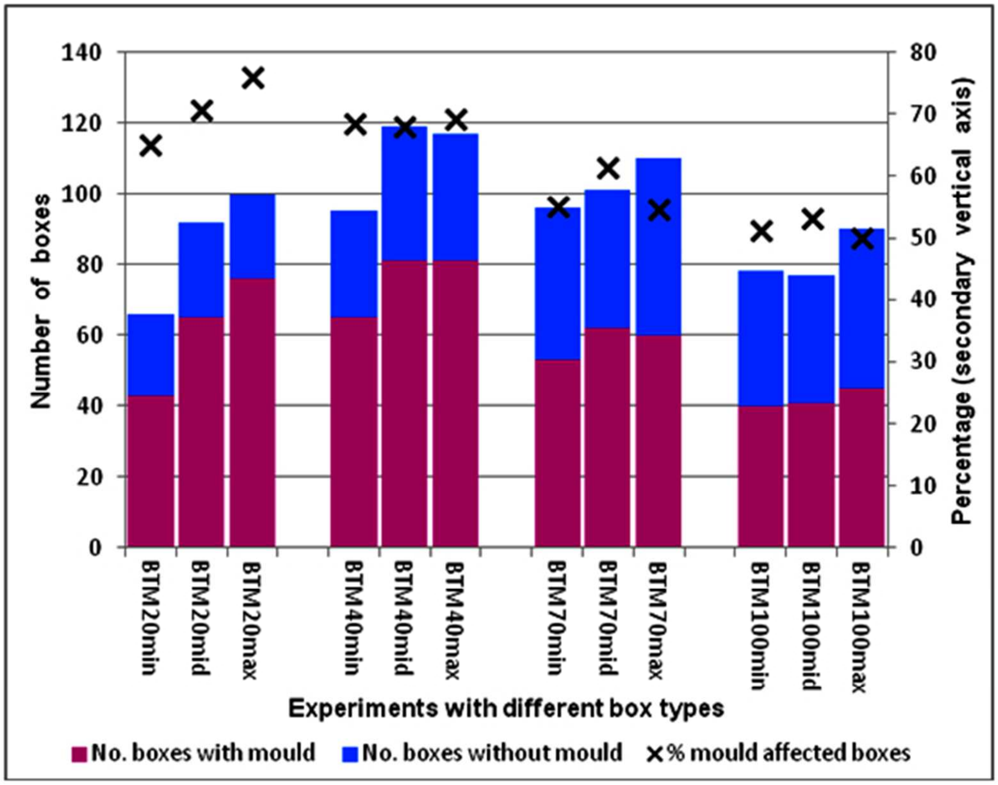


Figure 6: % mould affected boxes, grouped by BTM (left);

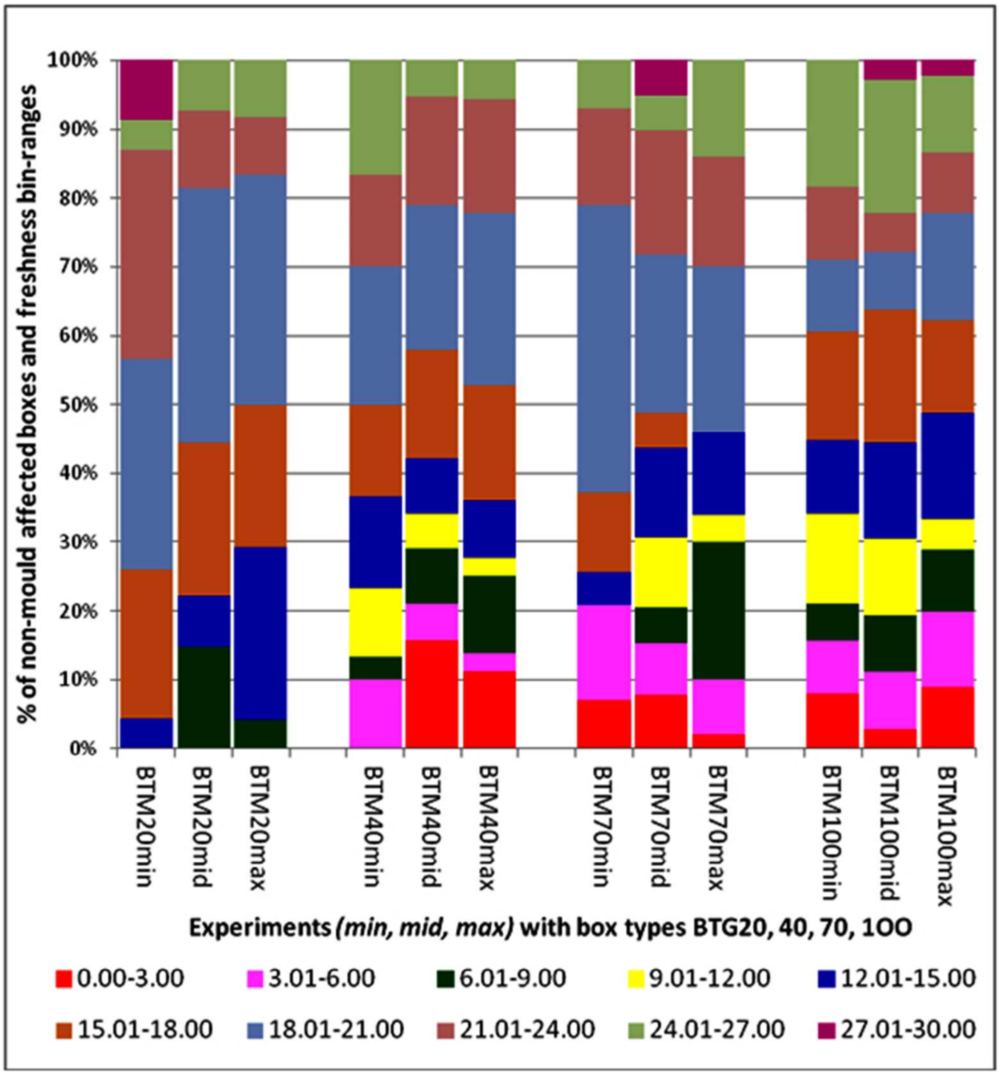


Figure 6: Histogram showing freshness % of boxes not affected by mould, grouped by BTM (right)

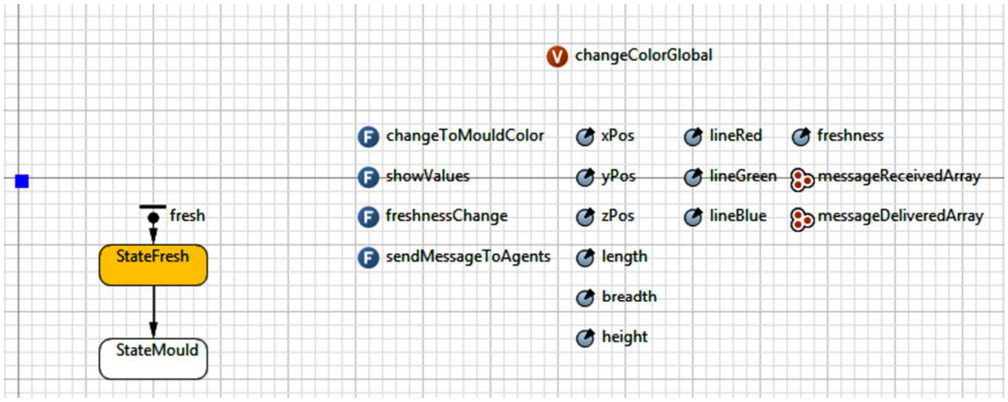


Figure 7: State chart