DECISION-MAKING FOR SELF-REPLICATING 3D PRINTED ROBOT SYSTEMS

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Andrew Burkhard Jones

In Partial Fulfillment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

Major Program:
Software Engineering

July 2020

Fargo, North Dakota

North Dakota State University
Graduate School

Title

Decision-Making for Self-Replicating 3D Printed Robot Systems

By

Andrew Burkhard Jones

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Dr. Jeremy Straub
Chair

Dr. Kendall Nygard

Dr. Simone Ludwig

Dr. John Bitzan

Approved:

7/31/2020 — Date

Dr. Kendall Nygard — Department Chair

ABSTRACT

This work addresses decision-making for robot systems that can self-replicate. With the advent of 3D printing technology, the development of self-replicating robot systems is more feasible to implement than it was previously. This opens the door to various opportunities in this area of robotics.

A major benefit of having robots that are able to make more robots is that the survivability of the multi-robot system increases dramatically. A single surviving robot that has the necessary capabilities to self-replicate could prospectively repopulate an entire 'colony' of robots, given sufficient resources and time. This gives robots an opportunity to take more risks in trying to accomplish an objective in missions where robots must be used instead of humans due to distance, environmental, safety and other concerns. Autonomy is key to maximizing the efficacy of this functionality (or allowing this functionality in a communication limited/denied environment) for this type of robotic system.

A challenge of analyzing self-replicating robot systems, and the decision-making algorithms for those systems, is that there isn't currently a standard means to simulate these systems. Thus, for the purpose of this work, a simulation system was developed to do just this. Experiments were conducted using this simulation system and the results are presented.

In this dissertation, the configuration and decision-making of self-replicating 3D printed robot systems are analyzed. First, an introduction to the concepts and topics is provided. Second, relevant background information is reviewed. Third, a simulation, used to model self-replicating robot systems to perform the experiments in later chapters, is detailed. Then, experiments are conducted utilizing this simulation model. These include the analysis of the impact of replication categories on system efficacy, the analysis of the comparative performance of multiple decision-

making algorithms, and cybersecurity threats for self-replicating robot systems. For each, data is presented and analyzed, and conclusions are drawn. Finally, this dissertation concludes with a summary of the results presented throughout the document and a discussion of the broader findings from the experiments.

ACKNOWLEDGEMENTS

Facilities, materials, and equipment used for this work have been supplied by the North Dakota Space Grant Consortium, North Dakota State University (NDSU) Department of Computer Science, NDSU College of Science and Math, NDSU College of Engineering, and the NDSU Foundation.

DEDICATION

This dissertation is dedicated to my parents, Joseph and Helga Jones, whose support has shaped who I am today. I also dedicate this work to my academic adviser, Dr. Jeremy Straub, who guided me in this process and helped me further my potential.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CHE..Centralized Heterogeneous.

CHO ....Centralized Homogeneous.

DHE ....Decentralized Heterogeneous.

DHO...Decentralized Homogeneous.

HHE ....Hierarchical Heterogeneous.

HHO...Hierarchical Homogeneous.

VAR ....Variable decision-making algorithm.

STR ....Strategic decision-making algorithm.

Std Dev...Standard Deviation.

FDM...Fused Deposition Modeling.

Num..Number of (in parameter names).

Env ....Environment (in parameter names).

Pr..Printable Components (in parameter names).

NonPr ....Nonprintable Components (in parameter names).

LIST OF APPENDIX TABLES

1. INTRODUCTION[1]

In this dissertation, the configuration and decision-making of self-replicating 3D printed robot systems are analyzed. First, an introduction to the topic is provided. Second, relevant background information is reviewed. Third, a simulation system developed to model self-replicating robot systems is detailed. Then, experiments are conducted which utilize this simulation system and the data from these experiments is presented and analyzed. These experiments include analysis of the impact of system configurations (as discussed in Section 1.2) on system efficacy, the analysis of the comparative performance of multiple decision-making algorithms, and the assessment of cybersecurity threats for self-replicating robot systems. Finally, this dissertation concludes with a summary of the results presented throughout the document and a discussion of the broader findings from the experiments.

1.1. Introduction

The concept of self-replicating robots has been around for some time—dating back to before the 1940s [1]. With the advent of 3D printing technology, the actual development of functional self-replicating robots is more feasible to implement than it was previously, opening the door to a multitude of research opportunities in this newly possible area of robotics.

A major benefit of having robots that are able to make more robots is that the survivability of the multi-robot system increases dramatically. A single surviving robot that has the necessary capabilities to self-replicate could prospectively repopulate an entire ‘colony’ of robots given sufficient resources and time. This gives robots an opportunity to take more risks in trying to

[1] This chapter is derived from previous work in: A. Jones and J. Straub, “Concepts for 3D Printing-Based Self-Replicating Robot Command and Coordination Techniques,” *Machines*, vol. 5, no. 2, Apr. 2017.

accomplish an objective in missions where robots must be used instead of humans due to distance, environmental, safety and other concerns. Autonomy is key to maximizing the efficacy of this functionality for this type of robotic system [2].

The remainder of this chapter is organized as follows. First, a categorization scheme for self-replicating robot systems is proposed. Second, the critical requirements for 3D printed self-replicating robots are outlined. Third, usable materials and acquisition of those materials by a robot system are discussed. Fourth, the decision-making criteria for self-replicating robot systems are detailed. Then, a conceptual prototype is presented. Finally, the aerospace applications of self-replicating robot systems are discussed.

1.2. Proposed Replication Categorization

In this subsection, the categorization scheme utilized for the experimentation is detailed. There are many attributes that could be used to characterize an overall replication scheme. For instance, whether multiple robots need to be involved in making a new robot, or if the hardware/design evolves over time [3] are both possible criteria. However, the categorization approach used for this work is to categorize the system based on what types of robots the system can build and which of those types have a replication-related capability. This (replication-related capability) means a capability used for fabricating, building, or assembling parts for a new robot.

The utilized categorization consists of a combination of two separate classifications The first classification, the replication approach, consists of centralized, decentralized, and hierarchical. The second classification, the production approach, consists of heterogeneous and homogeneous.

1.2.1. Centralized Replication Approach

With the centralized replication approach, robots that have a replication-related capability are not buildable by the robot system itself. Initial or factory-made robots are solely used for replication related capabilities. A replicator robot is setup and acts as a replication center, as depicted in Figure 1.1. It is able to build robots that don't have replication-related capabilities that can collect resources or accomplish other goals.

Having a central node dedicated to the replication process has benefits and drawbacks. One benefit to this approach is that the regular robots (produced by the central node) don't have to have the replication equipment installed, allowing the materials to be used for other purposes (i.e., it reduces material usage). However, a key drawback to this approach is that if the central replicator node malfunctions, it is a central point of system failure [2].

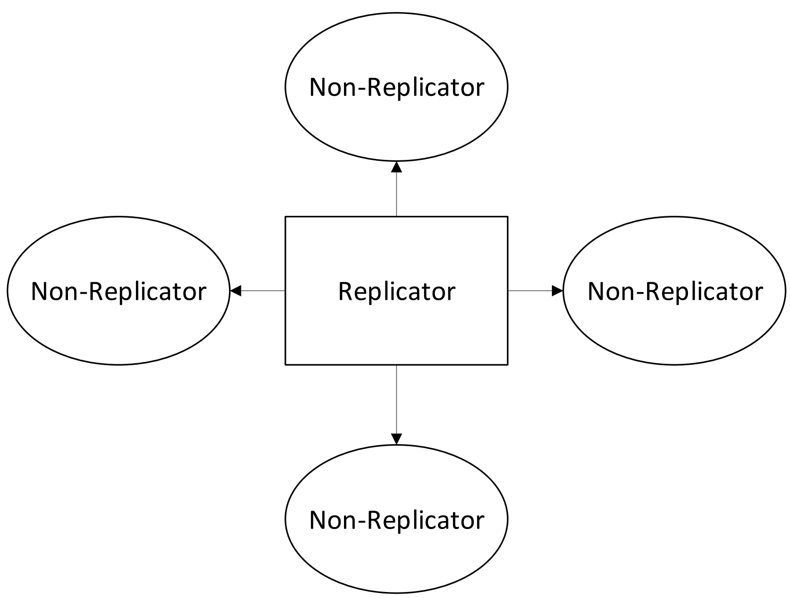


Figure 1.1. Centralized replication category.

Another benefit, for some applications, is that all of the replication-related materials collected are brought to a centralized stockpile. This prevents the replication process from getting bottlenecked by poor resource exchange between robots (although this could be potentially remedied by the implementation of a cooperation scheme in non-centralized systems) [2]. However, the requirement to bring all replication resources to the central node also presents drawbacks. This is particularly problematic when the resources needed for robotic production and the location needing the robots produced from these resources are located far away from the central production location.

1.2.2. Decentralized Replication Approach

The decentralized replication approach, which is depicted in Figure 1.2, includes replication capabilities in all of the robots in the robot system. A benefit to this approach is that, in the event of a catastrophic event, if one robot survives and has replication resources available, it could potentially make more robots to repopulate the system. This capability leads to increased survivability of the multi-robot system [2]. It may also allow the multi-robot system to split into multiple groups that do not have a dependency on a central hub (as they would in the centralized configuration). A drawback to this approach is that each robot needs to include replication equipment, which may be rare or specialized equipment. Even if it doesn’t require rare resources or equipment, the approach is likely to require additional standard resources. Another potential drawback is that a specialized replicator robot could potentially be made with higher quality equipment, which may make it more versatile in transforming raw materials into new robots or result in higher quality being produced. Not having the ability to use a wide array of raw materials may cause problems in certain environments.

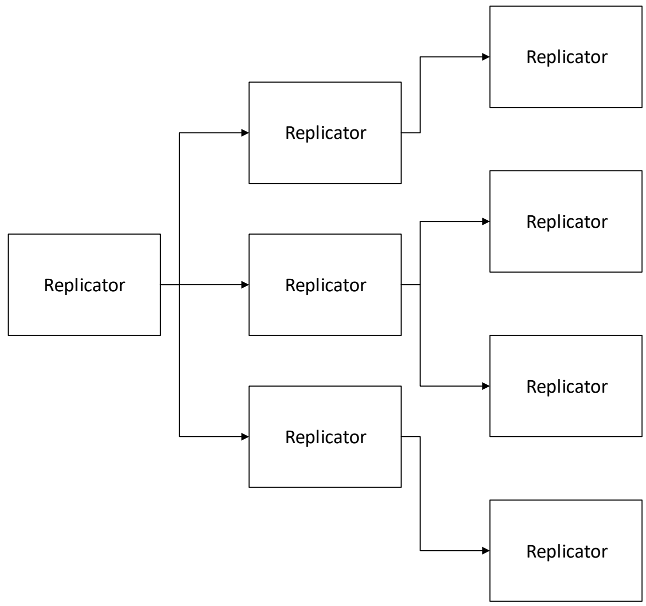


Figure 1.2. Decentralized replication category.

1.2.3. Hierarchical Replication Approach

The hierarchical replication approach, which is depicted in Figure 1.3., is a combination of the centralized and decentralized approaches. Systems of this type are capable of building robots with replication-related capabilities as well as robots without these capabilities. The benefits and drawbacks of this approach are dependent on the ratio of robots with replication-related capabilities to non-replicating robots. At the extremes, the benefits and drawbacks approximate those of the centralized or decentralized approach.

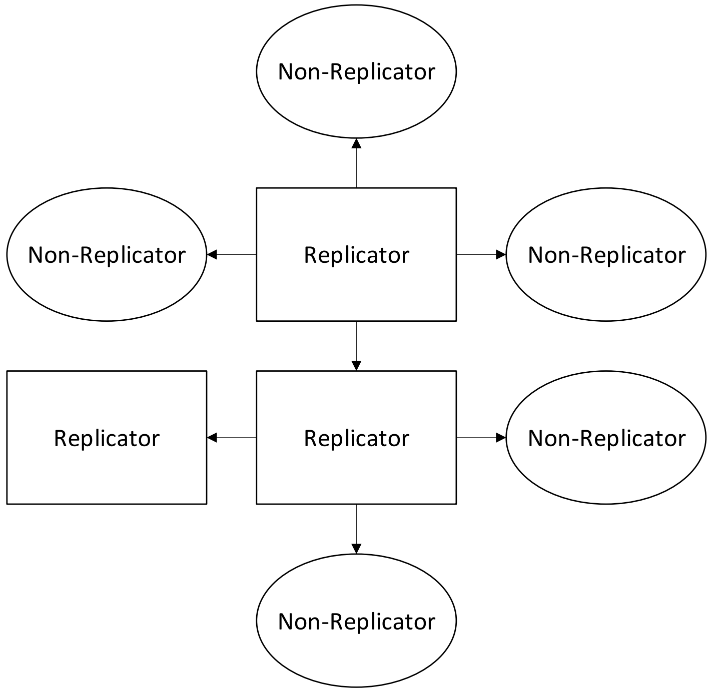


Figure 1.3. Hierarchical replication category.

1.2.4. Production Approach: Homogeneous or Heterogeneous

The production approach categorization is made in addition to (and is distinct from) the replication approach categorization. The production approach consists of two classifications, homogeneous and heterogeneous. For multi-robot systems, in general, a homogeneous system is composed of robots that are all of the same design, while having multiple types of robots makes the system heterogeneous [4]. Under the replication classification scheme used herein, the characterization of a multi-robot system as homogeneous or heterogeneous, in terms of replication, is as follows: the homogeneous production approach means that the system uses a single robot type that has all of the necessary replication-related capabilities. In contrast, the heterogeneous production approach means that the system uses two or more robot types that have replication-related capabilities. The replication-related capabilities used across the robot types used in the system must be partially independent of each other to qualify for this classification.

The benefits of a homogeneous system include the potential reduction in task allocation complexity, as only one robot is needed for replication tasks. This approach may also increase the survivability of the system as a single surviving robot could repopulate the robot system (given sufficient resources and time).

The benefits of a heterogeneous system include resource savings. Using the minimum amount of materials required to produce a functional robot—meeting relevant quality standards—is desirable. The heterogeneous approach allows the greatest number of robots possible to be constructed, given the available level of in-situ collected and stored materials. Tailoring robots to specific roles ensures that each robot only includes the necessary functionality, saving resources [5]. Another consideration is the selection of a design that makes use of resources that are available and abundant [2]. Choosing a design that utilizes locally abundant resources maximizes the number of robots that can be produced. A homogeneous approach would preclude this type of adaptation.

1.2.5. Parallels with Biological Organisms

The centralized replication approach has parallels with eusocial species. Eusociality is defined by the following characteristics: cooperative brood care, overlapping generations within a colony of adults, and a division of labor into reproductive and non-reproductive groups [6]. Some examples of species exhibiting eusociality include ants, termites, and certain species of bees [7]. In this regard, organisms with this social structure typically have a queen (also called a gyne) that gives birth to workers or other sterile castes. Thus, the reproduction is conceptually similar to the centralized approach. However, in the event of the queen dying or being otherwise indisposed, a member of certain other specialized castes can become the new queen (although this varies from species to species) [7]. Thus, the analogy is imperfect in that there isn't the central point of failure (perhaps if a species such as this existed, it may have gone extinct for this reason). The hierarchical approach has parallels to this as well in that there are groups that are reproductive and those that are sterile (although it would be a multiple queen dynamic). In contrast, the decentralized approach has parallels to organisms without sterile castes (most organisms). Furthermore, the categorization of homogeneous versus heterogeneous has parallels to asexual and sexual reproduction, respectively.

1.3. Requirements for 3D Printed Self-Replicating Robots

The capability to perform replication activities is a core functionality of a self-replicating robot system. The replication process can be performed in a single step or consist of multiple stages of construction, such as the fabrication of parts and their subsequent assembly. In this section, the required components for a self-replicating robot, based on 3D printing technology, are discussed.

1.3.1. 3D Printer

There are many different 3D printer designs and configurations that have been demonstrated. In terms of overall critical functionality, the necessary components of a 3D printer can be divided into four categories: material extrusion, cartesian movement, print bed, and control electronics.

1.3.1.1. Material Extrusion

The component of a 3D printer that carries out the additive manufacturing process is the extruder. The extruder consists of a material driving mechanism and a print head/nozzle [8]. In the relatively common 3D printing method of Fused Deposition Modeling (FDM), the print head is a thermal ‘hot end’ that forms molten beads from the polymer which is fed through by the driving mechanism. The solid filament being fed in acts as a piston, building pressure to force the molten polymer out of the nozzle [9].

1.3.1.2. Cartesian Movement

A 3D printer is a cartesian robot, or a machine that can move in three linear directions (x-, y-, and z-axes). Most 3D printer models use stepper motors to achieve this (one or more motors per axis), due to the high precision and accuracy required [8]. The stepper motors typically control the motion of the axes through timing belts or threaded rods. However, this is not the only design possible. For example, recently the Snappy RepRap design utilized a 3D printed gear system for the same purpose [10].

The cartesian movement mechanisms also benefit from, or may require, a frame that stabilizes the printer's axes. However, the design of a 3D printer incorporated in or mounted on a robot may instead choose to focus on combining the robot's frame with that of the 3D printer. This setup may make the printer more stable overall (due to complete structural integration). However, movement of the robot may cause the printer to shake or tilt.

1.3.1.3. Print Bed

The print bed is the surface which the 3D printer builds the printed object on. Feasibly, a print bed can be any reasonably level surface that can keep the printed object from moving while the print is in progress. However, this can prove challenging for various reasons – some of which are material dependent. While 3D printing materials must self-adhere and may adhere to similar materials, one of the key challenges in print bed design is getting the first layer to adhere to the print bed during the print and then being able to remove the printed object upon completion [8].

For FDM type 3D printers, the surface of the print bed is commonly made from either glass or aluminum in order to provide a cost effective smooth and level surface. In many cases this is covered by additional material. In addition, using a heating element to create a 'heated bed' works well with these materials because the heat spreads across the surface [8]. Some example types of covering materials that are used include painter's tape, Kapton tape, and PEI.

1.3.1.4. Control Electronics

Typically, the control electronics of a 3D printer consist of one or more microcontrollers and a main computer board [8]. In some models, these two components are combined into a single module. The microcontrollers run the software or firmware and controls the operations of the 3D printer. The main board interconnects all the components of the printer and distributes power to the individual components. It also relays commands to the components from the microcontroller.

1.3.2. Assembly Equipment

Work has been conducted on 3D printing articulated models without the need for a separate assembly system [11]. However, some sort of assembly capability may be well suited for allowing robots to print similarly sized robots. If the printing robot cannot produce and connect multiple small parts (or expand the printing area in some way), each subsequent generation of robots would be constrained to be smaller than the originals, so that they could be printed in the printing area [12].

Techniques for robotic assembly are well understood. Robotic assembly has been used for applications ranging from assembling small parts [13] to buildings [14]. Techniques for both independent and cooperative robotic assembly have been previously proposed [15], [16]. For example, a printing unit could be grouped with robotic arms (either on a single robot, or from a second cooperating robot). These arms could be used to assemble printed, stored, or otherwise obtained parts into their needed configuration.

1.3.3. Mobility

A requirement for self-replicating robot systems is the need for a capability that can be used for acquiring the needed materials to build new robots. These materials could be gathered in-situ, through robot foraging, or delivered by an outside source. Robot foraging is broadly defined as robots searching for and collecting objects, and subsequently bringing them to a collection point [17]. Robot mobility is necessary in order to collect resources and utilize them. The means of mobility that are appropriate for use depend on the operating environment, the expected available resources, and other considerations.

1.3.4. Communication

The coordination of a multi-robot system is heavily dependent on the ability of the robots to communicate. Communicating can be accomplished with various techniques and technologies. For instance, including radios in each robot would enable them to communicate at a distance, as well as removing the necessity to have a line of sight between them for communications (although obstacles and environmental factors could interfere with the signal). Alternatively, a coded visual system, such as blinking LED lights, could be used for relaying messages [2]. An even more range restricted approach would be to use physically attached wires that temporarily connect robots. Physical connections might be desirable for secure or high data transmission needs.

1.3.5. Processing

Each robot needs one or more computer units. Fabricating such a device in a factory setting with exact materials, machinery, and a relatively controlled environment is well understood. For self-replicating robots, especially those designed to forage for resources, growth is constrained by the availability of suitable materials, which would necessarily include either computer units or the raw materials to make them (if the fabricating robots possessed applicable fabrication capabilities)

[2]. Currently, processor fabrication uses large, heavy, and expensive equipment. This makes in-situ processor fabrication impractical. In the longer term, with suitable resources available in-situ, the development of a small, low cost processor fabrication capability may be possible. In the short term, however, processor availability is a limiting factor. It may be the primary determination of how many robots can be produced for a given mission, if computer units cannot be fabricated locally [2].

1.3.6. Sensors

Sensing of the environment is important for performing mission tasks, gathering resources, and navigating the terrain. Required sensing can include visual, audio, tactile, and magnetic capabilities. Robots may need to relay sensor data to other robots without sensing capabilities for some applications [2].

1.4. Material Usage and Acquisition

A self-replicating robot system inherently requires resources for robot fabrication, and (depending on mission objectives) would likely need a capability to acquire more. The availability of resources is a major constraint on the ability to assemble a new robot, as depicted in Figure 1.4. In this section, the materials that are suitable for 3D printing are discussed. Then, the means of acquiring resources by a robot system are reviewed.

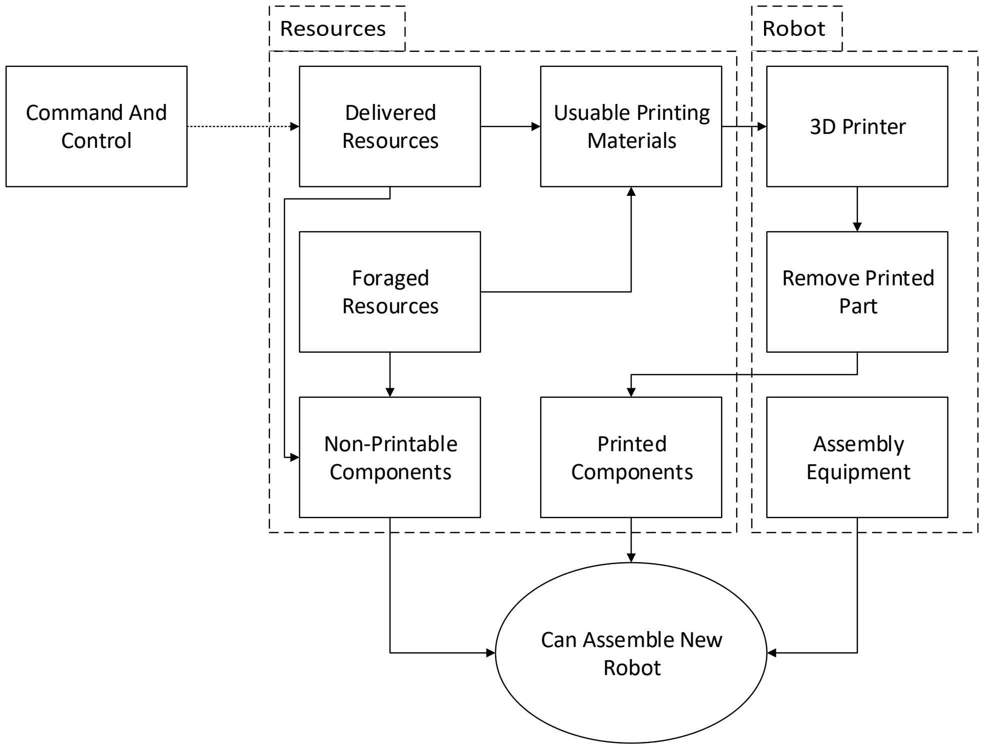


Figure 1.4. Robot assembly requirements.

1.4.1. Useable Materials

A planning requirement for fabricating a new robot is to have (or project having) the materials that are needed to build it. Various materials can be used in this process, with the constraints largely driven by the specific capabilities of the replication equipment. For robot systems operating at distant locations, the use of in-situ resources may be necessary to reach the desired mission duration and, potentially, facilitate having a greater ability to take risks while meeting other missions requirements and constraints [12]. Examples of in-situ resources being used for distant 3D printing include the use of basalt printing of structures for Martian exploration [18]. The use of a D-shape printer for building infrastructure out of regolith on the Earth's moon [19], and the use of a collection of simple self-replicating robots to exploit lunar material and energy resources [20], have also been previously proposed.

1.4.1.1. Printing Materials

An in-situ mission would likely not use current-generation commercial or consumer-grade 3D printing technology, however, this has been used for the purposes of experimentation. Most modern FDM 3D printers use filament of a specific diameter. Irregularities in the diameter of the filament can potentially cause jams and other problems [8]. This potential preparation is not desirable when considering the acquisition and potential preparation of suitable printing materials by a robot system, especially if using in-situ resource acquisition. An alternative FDM approach is the use of pellets in conjunction with a pellet compatible extruder. This would be similar to a foraging approach, as pellets are simply placed in a funnel to supply the extruder.

Other types of (non-FDM) 3D printing have different material requirements. Other 3D printing methods include the use of an ultraviolet laser to harden a photosensitive polymer (stereolithography), or using a laser to selectively melt metal or polymeric powder (laser sintering) [21].

1.4.1.2. Non-Printable Components

While some components may not currently be printable, this should be seen as topics for future research instead of a conceptual limitation. Advances in 3D printing technology continue to push the boundary of what is printable. For instance, conductive filament has been used to print simple circuits [22]. Certain types of actuators, such as soft dielectric elastomer actuators, have also been 3D printed [23].

Components that cannot be 3D printed would need to be supplied to the robot system or fabricated using another method. For instance, milling (subtractive manufacturing) [24] capabilities could be used in conjunction with 3D printing, in order to work with certain metals and other materials that cannot be 3D printed. Another potential benefit of milling includes rectifying issues detected on a printed object or component.

1.4.2. Acquiring Materials

A constraint for self-replicating robot systems is the means of acquiring the needed materials to build a new robot. These materials could be gathered in-situ through robot foraging or delivered by an outside source [12].

1.4.2.1. Robot Foraging

Robot foraging is broadly defined as robots searching for and collecting objects, and subsequently bringing them to a collection point [17]. This can be divided into multiple challenges; this includes the need for techniques for recognizing resources and collecting them.

Recognizing resources in a given environment can be accomplished in many ways. For example, reflection seismology (similar in concept to radar) has been used to discover oil and natural gas [25]. Magnetic surveys can used to detect ore deposits [26]. The visual recognition of surface resources can be accomplished by processing images using trained deep convolutional neural networks [27]. The foregoing techniques could be used to identify many of the resources required for robot replication. Alternate techniques may be needed for robots with additional resource identification needs.

Once resources are identified, the robots must have the capability to autonomously collect them. A number of examples of prior work demonstrate relevant capabilities. For example, Green and Vogt [28] proposed a multi-robot system that could cooperatively and autonomously mine ore using rock drills. Dunker, et al. [29] demonstrated a proof of concept for utilizing teams of robots that could automatically gather regolith on the surface of the Moon with an actuated scoop and bring it to a central processing station. Each of the foregoing is a capability that could be integrated into the self-replication robots' design to provide basic functionality. As with resource identification, the resource collection capabilities of robots (both in terms of physical collection hardware and commanding software) may need to be augmented to support mission-specific collection requirements [2].

1.4.2.2. Delivered Resources

Resources that cannot be printed or otherwise made from in-situ materials may need to be delivered from an outside source. A special case of delivered resources is the initial deployment of the robot system. The initial resources would include the initial robot(s) and, potentially, a supply of important non-fabricable components (i.e., processors for subsequent robots) [5].

1.5. Decision to Replicate

The decision of when a self-replicating robot system should replicate is affected by several factors. These factors include the available resources, available replication equipment, current objectives, and robot capacity. These factors are depicted in Figure 1.5 and are discussed in the following subsections.

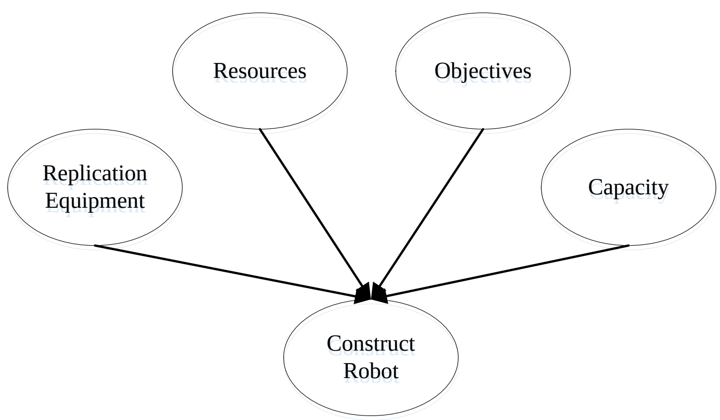


Figure 1.5. Key decision-making factors for the new robot construction decision.

1.5.1. Available Resources

Resource availability is critical to the decision to build a robot. Multiple factors, as shown in Figure 1.6, contribute to the resource availability characterization that is supplied to the decision-making algorithm. A requirement for fabricating a new robot is to have (or project having, at the requisite time) the materials that are needed to build it. Another factor to consider is the quality or purity of the resources. Poor quality resources could impact the quality of the finished product. This (ideally) should be considered when deciding whether to move forward with the replication process.

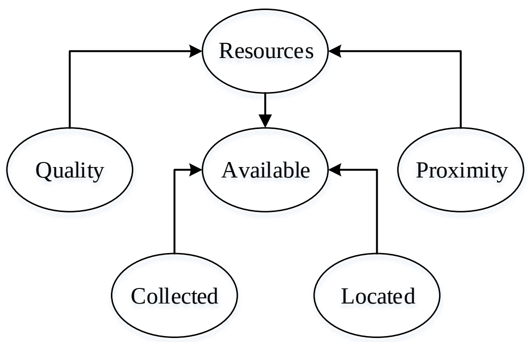


Figure 1.6. Mission resource characterization.

1.5.2. Replication Equipment

The replication equipment present needs to have the capability to produce the needed design, otherwise the process cannot move forward. There is a probability of success associated with a given unit of equipment's capability to perform a given printing task. If this value is known, it can be factored into the decision-making process. This probability value can be estimated based on factors such as known wear or damage to the equipment and the performance of the equipment in pristine condition [2]. This value can also be determined empirically by measuring the results of production and comparing them to the expected results [30]. Even if a given equipment configuration can build a desired design, it may have known limitations and certainly has the potential to encounter errors. Issues can include the equipment jamming, printed parts not fitting together properly, and adverse environmental conditions.

1.5.3. Objectives

The current mission objectives, as shown in Figure 1.7, are relevant to the decision to build a new robot (or not). These objectives determine the necessity for fabricating a new robot, and what design it should have. This information would drive the need (if required) for increasing the quantity of robots or optimizing a design for a specific capability.

An increase in the quantity of robots may be needed for certain exploration efforts, or to support planned future robot fabrication predictions [2]. An alternative consideration is that a new robot of a particular design may be needed for a specific task that has a necessary benefit, such as reaching and collecting a resource that is out of reach of the current robots in the system.

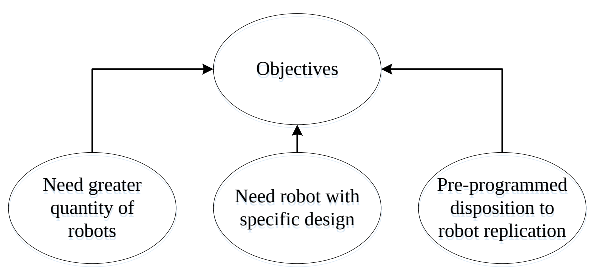


Figure 1.7. Generic objectives for robot creation.

1.5.4. Capacity

Depending on user choices or implementation and design constraints, it may be that the robot system is capped at having a certain maximum number of robots. The system capacity factor, depicted in Figure 1.8, characterizes the ability of the system to support more robots, to inform the build-or-not decision-making process [2]. One example of a restriction is having limited centralized command and communication capabilities, such that the number of robots that the central robot can command or communicate with is limited. Another restriction would be the need for a certain resource to continue functioning over time, such as energy or replacement parts. Finally, the number of robots that a system can have may also be restricted by the space available in the operating environment. Small spaces would necessitate having a fewer number of robots for optimal performance. This notion of an optimal number of robots as opposed to a maximum is also a more general consideration.

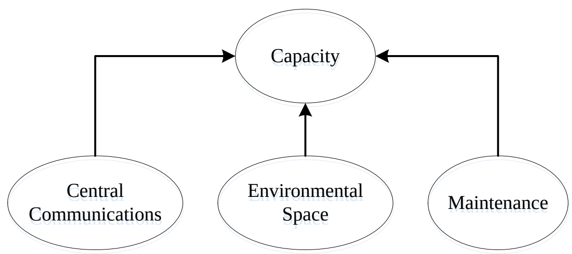


Figure 1.8. Example constraints on the maximum number of robots needed.

1.5.5. Example Implementation of System Operations

An example of the operations of a self-replicating robot system is now considered. Figure 1.9 depicts the general decision-making process undertaken by each robot in the example system. This diagram assumes a limited set of objectives, including repairing robots, building robots, idling, gathering resources, exploring, and performing other mission-related objectives.

One point of interest is the flow of locating resources, to collecting resources, to having them available to use as materials (for new robots or for repairs to existing ones). The ‘explore’ objective locates resources and contributes to the terrain map. Located resources can subsequently be gathered by a robot that is pursuing the ‘gather resources’ objective. Gathering resources contributes to the available resources, which may then be used to build or repair robots. These different steps can be performed by the same robot or performed by different robots, depending on robot capabilities and decision making. Communication amongst the robots in the system updates relevant databases to reflect changes in resource status. This is important because this data is used as part of the decision-making process, such as to decide whether the system carries out a task or not. It is also used to assign robots tasks, based on the task allocation scheme. For example, a construction robot could be tasked with updating data regarding resources available in the resource pool to indicate that certain materials are no longer available, as they are used to construct or repair a robot.

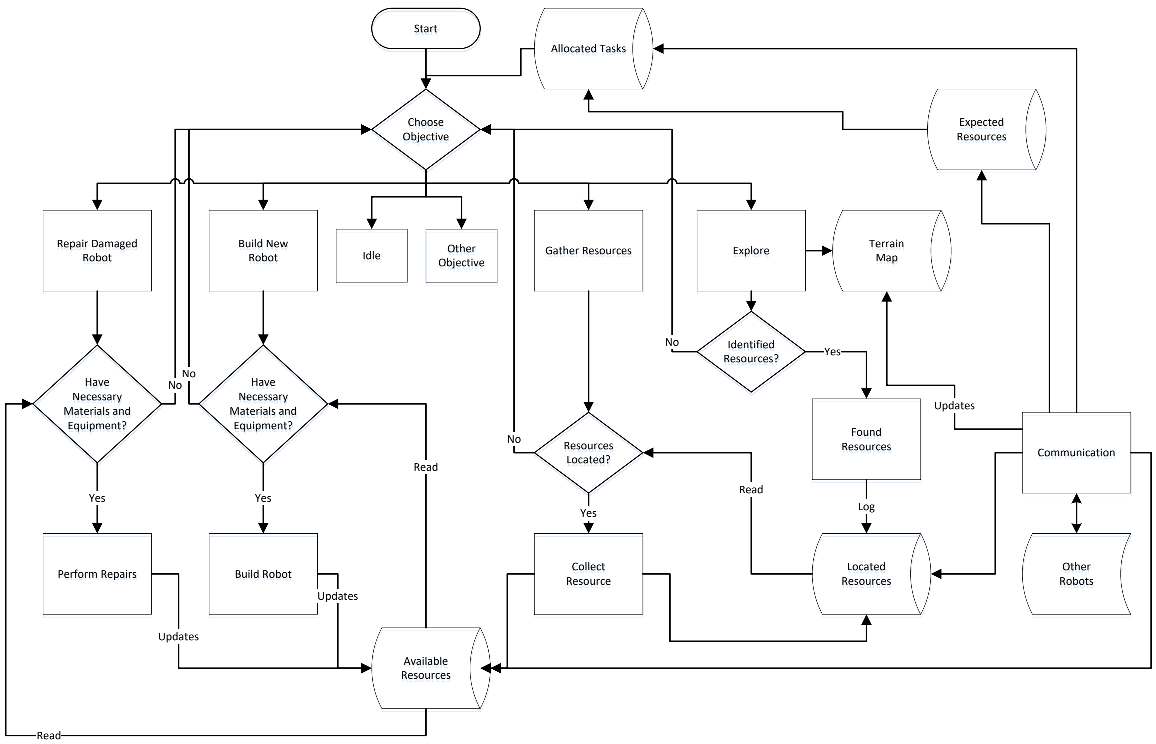


Figure 1.9. Example system decision-making diagram.

1.6. Conceptual Prototype

A conceptual example robot system, a (mostly) 3D printed 3D printer, was built (depicted in Figure 1.10). Its design is a slightly modified version of the “Snappy” RepRap [10]. Its frame is 3D printed and ‘snaps’ together using the mechanical design of the printed components. The nonprintable components include stepper motors (for moving parts), a heating block and an extruder, a heated print bed (for print adherence), and control electronics. It is capable of 3D printing components that could be used to assemble an additional 3D printer or robot. However, the prototype system does not include an automated assembly capability.

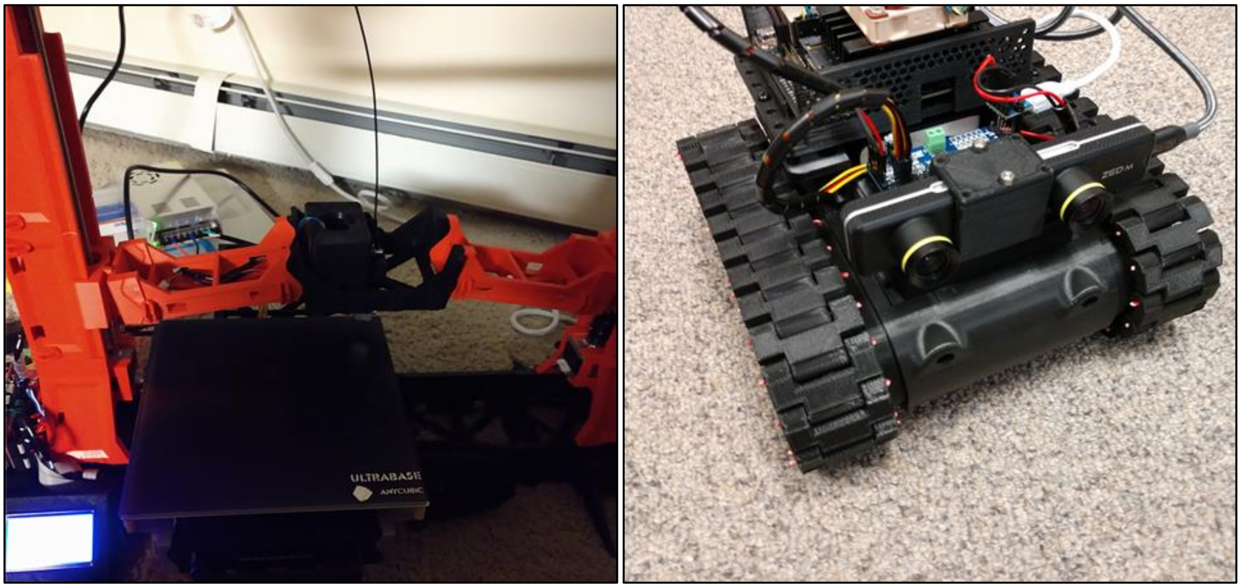


Figure 1.10. 3D printed 3D printer (left). 3D printed 'normal' robot (right).

In addition, a small (mostly) 3D printed robot was constructed and programmed with software that allows it to perform simple resource location tasks (i.e. find a colored spool of 3D printer filament). This small 3D printed robot is meant to compliment the 3D printed 3D printer for the purposes of the example implementation. It consists of a 3D printed chassis and 3D printed tracks which are driven by two servo motors (based on design from [31]). Its on-board computer is a Nvidia Jetson Nano, which processes the visual data from the Zed Mini stereo-camera in order to perform simple resource identification and localization. It is powered by a rechargeable lithium ion battery pack that fits within the chassis.

1.7. Aerospace Applications

The application areas for self-replicating robot systems include areas that are difficult for humans to access or where it is prohibitive to bring the materials and supplies required for crewed missions. This type of robot system could theoretically be used in a wide variety of applications.

An application domain that may especially benefit from self-replicating robots is space exploration. Launching materials into space can be prohibitively expensive, which may drive a need for utilizing in-situ materials [12]. In this section, example aerospace applications that may benefit from the use of self-replicating robots are discussed.

1.7.1. Planetary Exploration

A self-replicating robot system could be used for planetary exploration. This would be especially useful due to the high cost of transporting materials to remote planets. Constructing additional robots could facilitate a faster exploration rate, compared to a fixed number of robots. Furthermore, a single surviving robot that is able to self-replicate could repopulate an entire 'colony' of robots with sufficient resources and time. This would give the robots an opportunity to take more risks in trying to accomplish an objective. However, the usefulness of this approach would largely depend on the presence of in-situ resources and how usable those resources are in terms of constructing additional robots. For example, basalt 3D printing could be useful on Mars, due to its abundance [18].

1.7.2. Satellites

Three-dimensional printed self-replicating satellite systems could be used for a variety of applications. One of the associated benefits with this type of system is that constructing a structure or craft in microgravity has the benefit of it only having to be structurally designed for microgravity (instead of terrestrial gravity and launch forces) [5]. Some estimates suggest that roughly 30% of a spacecraft's structural mass could be removed if that craft were built in space rather than on Earth [32]. However, acquiring materials in space can pose a problem in most cases, so materials would either need to be supplied or mined from planetary body surfaces.

One specific example is that a self-replicating robot/satellite system could be used to mine asteroids. The use of this type of system for mining missions may also solve the problem of supplying the replication materials needed to construct new craft. In this case, regolith and powdered metal from the asteroids could potentially be used to 3D print components for new craft [33]. For energy, solar power could be utilized in situations where it is available in sufficient amounts (i.e., at an appropriate distance from the sun and with few obstructions to sunlight). To this end, certain materials harvested from asteroids could potentially be used to construct portions of solar arrays [34]. Alternative sources of power could come from harvested resources, such as water (which could also be used as a propellant [33]).

1.8. Conclusion

In this chapter, an introduction to the concepts and topics of this work was provided. In Chapter 2, relevant background information is reviewed. In Chapter 3, a simulation, used to model self-replicating robot systems to perform the experiments in later chapters, is detailed. Then, experiments are conducted utilizing this simulation model. These include the analysis of the impact of system categories on system efficacy (Chapter 4), the analysis of the comparative performance of multiple decision-making algorithms (Chapter 5), and cybersecurity threats for self-replicating robot systems (Chapter 6). For each, data is presented and analyzed, and conclusions are drawn. Finally, this dissertation concludes with a summary of the results presented throughout the document and a discussion of the broader findings from the experiments (Chapter 7).

2. BACKGROUND

In this chapter, a review of relevant prior work is provided. First, information on self-replicating robots and related fields is discussed. Second, the parallels between self-replicating machines and biological organisms is presented. Last, certain relevant aspects of robotic artificial intelligence is detailed.

2.1. Self-Replicating Robots

In this section, the research topics of self-replication, self-assembly, additive manufacturing, in-situ material usage, and soft robots are discussed.

2.1.1. Self-Replication

Von Neumann is seen by many as the father of self-replicating machines [2]. In the 1940s, he investigated the logical foundations of self-replication [1]. In the 1950s, he proposed a self-replicating structure [35]. This work was presented in Scientific American [36], bringing it into the consciousness of the general public. After von Neumann's death [35], Burks completed his design for a 29-state automaton and published it in 1966 [37]. More recently, the technology to actually perform self-replication has become feasible. Beuchat and Haenni [35], for example, created a hardware implementation of cellular automaton and published the results of its analysis in 2000.

In [38], Lee, Moses, and Chirikjian follow a von Neumann-inspired framework and, in this context, define the degree of self-replication and task complexity. Self-replication complexity is presented in terms of an equation that compares the complexity among subsystems and overall system complexity. Mathematically, this can be presented as:

$$D_s = \frac{C_{min}}{C_{max}} * \frac{C_{total}}{C_{ave}} * \frac{1}{C_{ave}} \quad \text{(Eq. 1)}$$

Where C_{min} is the module under test with the least complexity and C_{max} is the module with the most complexity. C_{total} and C_{avg} present the sum and mathematical mean, respectively. Entropy (based on Sanderson's model [39]) is used as a measure of task complexity.

In addition to logical and theoretical works, a number of efforts have been made to create hardware systems. Suthakorn, Kwon, and Chirikjian [40], for example, demonstrated the operations of a semi-autonomous robot made from LEGO Mindstorm kit parts, that performs supervised replication. In [40], they built upon previous work (in [41]), where a concept and initial remote-controlled replication were presented. In [42], machine vision and other capabilities were added to the system, to facilitate autonomous operations. Similarly, Zykov, et al. [43] demonstrated real-world replication using modular robots based on specially-produced cubes. These robots collect cubes from feeder troughs and use them to produce equivalent copies. Even more flexible is the work presented in [44], where an algorithm for duplicating shapes using 'smart sand' replicates the shape of presented 3D objects. This work's efficacy was demonstrated via hundreds of simulated test runs.

More practically, the RepRap 3D printer [45] has been used as a template for the creation of numerous consumer-grade 3D printers. A RepRap printer is based—in part—on the use of parts printed on another 3D printer. Once a user has a working RepRap printer, he or she can produce many of the mechanical parts required to make another RepRap printer for his or her own use or for use by another person. Unlike the systems proposed herein, which use autonomous replication, RepRap printer construction requires significant human involvement.

2.1.2. Self-Assembly

Self-assembly is the autonomous organization of components into patterns or structures without human intervention [5][46]. In robotics, this can either refer to the ability of a kinematic machine to manipulate a series of parts into an assembled copy of itself [45] or the capability of a group of mobile robots to autonomously connect to and disconnect from each other (modular robotics) [47]. This differs from self-replication in that the resulting system is not necessarily capable of making, catalyzing, or in some way inducing more copies of itself [48].

Von Neumann proposed a self-assembler (that used a cache of spare parts) [1]. Whitesides and Grzybowski [46] demonstrated the similarity of natural and mechanized self-assembly. Butler, Murata, and Rus [49] advanced mechanized self-assembly by developing algorithms for a generic self-reconfiguring robot to divide and reform into different configurations. Sahin, et al. [50] demonstrated how self-assembly and disassembly can be used to allow a swarm of robots to collaborate, in some instances, while retaining the capability to perform tasks independently and in smaller groups, in other instances. Cooperation [47] and colonies [51] of self-assembling robots have also been proposed.

2.1.3. Additive Manufacturing

Additive manufacturing (AM) is a process of joining materials to make objects from 3D model data, usually building by printing layer upon layer [52][53]. AM methodologies, such as 3D printing, have been used for various commercial purposes. These application areas include fabricating prototypes, replacement parts, automobile components, aircraft components, robotic components, hearing aid molds, dental crowns, eyeglass frames, and prosthetic limbs [21][54][55].

A common AM method, widely used in modern commercial 3D printing, is fused deposition modeling (FDM), which is also referred to as fused filament fabrication. This method involves extruding polymer through heated nozzles to create a part’s cross sections [9]. Various other AM methods are used in industry, including the use of an ultraviolet laser to harden a photosensitive polymer (stereolithography), or using a laser to selectively melt metal or polymeric powder (laser sintering) [21].

2.1.4. Robot 3D Printing and the Use of In-Situ Resources

Creating robots with 3D printing has been demonstrated numerous times [2]. A variety of robots and robot parts have been created using 3D printing. The MU-L8 robot [56], for example, utilized 3D printed limbs to emulate human movements and play robot soccer. For more distant applications, the use of in-situ resources is necessary to prolong mission duration and, potentially, facilitates having a greater ability to take risks.

Examples of in-situ resources being used for 3D printing include the use of basalt printing of structures for Martian exploration [18]. The use of a D-shape printer for building infrastructure out of regolith on the Earth's moon [19], and the use of a collection of simple self-replicating robots to exploit lunar material and energy resources [20], have also been previously proposed.

2.1.5. Soft Robots

Soft Robotics refers to robotic devices that are fabricated from soft, flexible, materials, instead of the hard plastics and metals traditionally used in robotics [23], [57]. Soft robots could be beneficial for self-replicating systems since they may require fewer nonprintable components, due to less reliance on motors. An overall trend is that they tend to trade precision and deterministic control for bioinspired compliance and physical robustness [58]. An overview of how they have been designed, fabricated, and controlled is presented in [59].

Hiller and Lipson [58] demonstrated the automatic design of freeform soft robots for forward locomotion, using soft volumetrically expanding actuator materials. In [57], "smart materials" (materials which change their physical properties in response to external stimuli) were demonstrated and used to create a tentacle-like active structure which was employed for movement. In [60], Bartlett, et al. employed multi-material 3D printing to manufacture a combustion powered robot whose body transitions from a rigid core to a soft exterior. The robot is powered by the combustion of Butane and Oxygen and can perform untethered jumping.

2.2. Parallels with Biological Organisms

It is beneficial to consider design concepts based on nature. In [61], Pfeifer, Lungarella, and Lida present a discussion of how robots will eventually be able to exhibit certain desirable properties of biological organisms, such as adaptivity, robustness, versatility, and agility. Biological organisms also provide a model for replication, reproducing in many ways and adapting to their environment through the process of natural selection [62]. Similar to this, self-replicating robots could build other robots that are better suited to the environment or to objective-related needs [2].

The behavior of certain species of organisms can also provide design inspiration for robots. For instance, certain social animals, such as ants and birds, exhibit intelligent collective behavior. Observations of these animals provided inspiration for swarm intelligence [63].

However, while biology provides many insights, certain design considerations cannot be directly inferred by studying organic life. For instance, robots do not share certain constraints that animals have, such as the need to maintain a running metabolism. Certain technological solutions may also be superior to natural counterparts. Pfeifer, et al. [61], for example, suggested that this was the case for the wheel.

In terms of future applications, Dickinson [64] speculates that, as mechanical capabilities increase and are able to implement such designs, engineers may adopt more and more design concepts from nature. In this section, the topics of organic materials and self-perpetuating systems are discussed.

2.2.1. Organic Materials

Organic materials can prospectively be used as part of a robot's electronic system [2]. For instance, organic semiconductors such as organic thin film transistors (OTFT) may be able to be used as printable low-cost materials for a wide variety of applications [65]. This may make producing robots from in-situ resources easier to accomplish and increase possible fabrication options so that specific resource scarcity may become less burdensome. Furthermore, it may be possible to grow the organic compounds necessary to craft organic electronics [65].

2.2.2. Self-Perpetuating Systems

Self-perpetuation refers to the capability of something to cause itself to continue to exist. Self-replication is a subset of self-perpetuation. Self-perpetuating systems present special challenges which have been documented in numerous fields beyond robotics. Numerous examples of such systems exist within nature.

In biology, cell signaling and feedback mechanisms have been shown to be able to enter non-reversible states with either a positive or double negative feedback loop [66]. Platelets have been shown to have a potential role in a similar signaling self-perpetuating loop that may be associated with acute coronary syndrome [67]. A connection has also been suggested between in utero processes, which become self-perpetuating, and hypertension later in life [68] and self-perpetuating brain protein truncation leading to Alzheimer's, Parkinson's and Huntington's diseases [69]. Duchenne muscular dystrophy has also been suggested to be tied to a self-perpetuating feedback mechanism [70]. A collagen is produced that inhibits the regeneration of muscles and leads to more collagen production. A similar self-perpetuating cycle was demonstrated with prion protein conversion that can be triggered by proteasome inhibitors and potentially spread to brain tissue [71]. Self-perpetuation has also been suggested as a potential cause for human autoimmune disease [72].

In geology, faults in atomic configuration stacking have been demonstrated to create self-perpetuating steps in crystals [73]. In astrophysics, a self-perpetuating coating with a catalytic has been suggested as a reason for macromolecular carbon in some meteorites [74]. Self-perpetration has also been demonstrated at a much larger scale with D'Onghia, Vogelsberger and Hernquist [75] demonstrating these properties for disk galaxies' spiral-shaped arms.

Self-perpetuation is clearly present extensively throughout nature. Key to this is the creation of successive generations by the current generation. When a similar concept exists in computing, cybersecurity or robotics, these natural properties may be instructive in assessing its impact and implications.

2.3. Robotics and Artificial Intelligence

In this section, the topics of robot autonomy, robot foraging, manufacturing automation, multi-robot coordination, and swarm robotic control are discussed.

2.3.1. Robot Autonomy

Robot autonomy can be described as a system's capability to carry out its own processes and operations [76]. Further criteria may include a robot's sensing, decisional and actuation capacities, as well as its behavior in regard to its current environment [77]. Over the years, robots have been developed with varying levels of autonomy – depending upon application requirements and the technology available [5]. An example is the development of self-driving cars (also known as autonomous vehicles), which has progressed at an unanticipated pace in recent years [78].

For self-replicating robots, autonomy will need to factor in the use of multi-robot coordination (even for a system starting with a single robot, the replication process would make it a multi-robot system). Considerations for this include using centralized or decentralized decision-making, task and motion planning, and resource conflict resolution techniques [4]. One example coordination system, developed by Xu, et al. [79] was the Adaptive Parameter EXploration (APEX) algorithm. This algorithm is capable of adapting an arbitrary robot system to dynamic changes in task objectives and conditions during a session.

2.3.2. Robot Foraging

Robot foraging is broadly defined as robots searching for and collecting objects, and subsequently bringing them to a collection point [17]. This can be divided into multiple separate stages, such as the problem of recognizing resources and the problem of collecting them [2].

Recognizing resources in a given environment can be accomplished in many ways. For example, reflection seismology (similar in concept to radar) has been used to discover oil and natural gas [25]. The visual recognition of surface resources can be accomplished by processing images using trained deep convolutional neural networks [27]. Magnetic surveys can used to detect ore deposits [26]. The foregoing techniques could identify many of the resources required for robot replication. Additional techniques may be needed for robots with additional resource identification needs [2].

Once appropriate resources are identified, the robots must be able to autonomously collect them. A number of prior experiments and applications demonstrate relevant capabilities. For example, Green and Vogt [28] proposed a multi-robot system that could cooperatively and autonomously mine ore using rock drills. Similarly, Shaffer and Stentz [80] tested a robotic system for coal mining that could autonomously navigate and reposition itself underground using a laser range finder. Dunker, et al. [29] demonstrated a proof of concept for utilizing teams of robots that could automatically gather regolith on the surface of the Moon with an actuated scoop. It would then bring the collected materials to a central processing station.

For the coordination aspect of foraging, Baldassano and Leonard [81] described measures of performance that can be used to allocate tasks amongst multiple robots in a system. Cai [82] developed a learning algorithm to handle foraging tasks in completely unknown environments. Some coordination strategies have drawn inspiration from nature, such as work done by Fibla and Bernardet [83]. This work demonstrated a strategy for robot foraging based on the behavior of rodents. Similarly, Hecker, Carmichael, and Moses [84] described a resource cluster prediction algorithm, inspired by ant foraging behavior, that exploited the natural clustering of resources to efficiently direct robots to find and collect them.

2.3.3. Manufacturing Automation

Manufacturing automation uses electrical-, mechanical-, and computer-based solutions to operate and control a production process [85]. It is becoming more popular as markets drive rapid product enhancements and the customization of products requires the use of flexible automation infrastructures [86]. To this end, Saliba, Zammit, and Azzopardi [87] presented a strategy and proposed a set of practical guidelines for reconfigurable manufacturing automation.

For a machine to be self-replicating, it must be able to automatically manufacture a replica of itself. Collaborative robots that are used in manufacturing plants are becoming more flexible and efficient [88]. Robots are now considered as an integral part of some industries, due to their role in improving accuracy, repeatability reliability, preciseness, and efficiency [85].

The software that manages the automation process, referred to as manufacturing automation software projects (MASP), includes information regarding applied automation hardware and is becoming more complex. In [89], an approach for model driven development of automation software, based on the systems modeling language, is discussed. In addition, Vyatkin [90] provides an overview of state-of-the-art software engineering for industrial automation.

The increasing capabilities (hardware and software) created for manufacturing automation make a significant contribution to work on self-replicating robotics [2]. The automation of the replication process is a necessity for robots to be truly self-replicating.

2.3.4. Multi-Robot Coordination

The study of multi-robot coordinated systems, according to Yan, Jouandeau, and Cherif [4], has recently increased “significantly in size and importance”. They attribute this to the resolution of many previously vexing issues in single robot systems, as well as to specific multi-robot system needs. A number of key decisions define the coordination of a multi-robot system [4]. These include decisions related to the use of static or dynamic coordination, explicit or implicit communications, cooperative or competitive approaches, and centralized or decentralized decision-making. Task and motion planning and resource conflict resolution techniques also need to be identified.

Several examples of coordination approaches exist. Nieto-Granda, Rogers, and Christensen [91], for example, compared three exploring and mapping strategies: the reserves, divide and conquer, and buddy system approaches. Under the reserves approach, extra robots wait in the starting area until they are needed and are then given tasks. Under the divide and conquer approach, robots travel in as large of a group as possible and split in half when new navigation goals are uncovered. Finally, with the buddy system approach, robots travel in teams of two, until new navigation goals are detected. Similar to the divide and conquer approach, the team will then split, following both paths.

Portugal and Rocha [92] compared two techniques for commanding a multi-robot system used for the patrolling of a given area. The first technique seeks to optimize local gain (using a greedy Bayesian strategy). The other technique seeks to reduce interference and foster scalability (using a state exchange Bayesian strategy). They found that both approaches sufficiently solve the problem; however, the state exchange strategy outperformed the greedy strategy.

A wide number of examples of multi-robot coordination use exists. Liu, et al. [93], for example, presented a control system for a collection of life science laboratory mobile robots. Pennisi, et al. [94] presented the use of multi-robot surveillance (for indoor public places) using a distributed sensor network that combines RFID tags, mobile robots, and RGBD cameras. Starke, et al. [95] demonstrated close-proximity multi-robot operations for a welding automation application.

To meet the challenges presented by distributed systems, a variety of approaches have been suggested. Caliskanelli, Broecker, and Tuyls [96] presented a swarm-inspired method (based on the pheromone signaling behavior of honey bees), called BeePCo, to maximize the total area covered by a robot network in an environment. Swarm control styles are discussed in greater detail in the subsequent subsection. Straub [97] proposed a boundary node-based Blackboard Architecture approach for limiting the data replication traffic, to facilitate local robot decision-making. Jullian, et al. [98] proposed an information theoretic approach that iteratively estimates the environment state using a sequential Bayesian filter and a gradient of mutual information for the purposes of distributed control. Jin, XingJie, and ZengRong [99] explored the use of robot coordinated adaptive tracking. They presented a control algorithm with the distinctive feature that only a subset of followers would need to access the position information of a dynamic leader in the task space, reducing communications and other resource needs.

2.3.5. Swarm Robotic Control

With swarm robotic control, simplistic local rules are used to create complex behaviors [100]. This approach is patterned on insect colonies where groups of insects perform behaviors that are too complicated to be coordinated by any one insect's capabilities [100]. Sahin [101] proffers that swarm robotics involves the use of a "large number of relatively simple physically embodied agents" from which a "desired collective behavior emerges from the local interactions among agents and between the agents and the environment". Practically, this means that members of the robotic system can have simplistic command software and reduced processing capabilities, but still produce a complex outcome. Several efforts to classify swarm robotic systems have been performed. Abukhalil, Patil, and Sobh [102] define four high-level categories for robotic systems: swarm, self-replicating, self-reconfigurable, and modular. Significant overlap between these categories exists. Groβ, Dorigo, and Yamakita [51], for example, combined self-assembly/reconfiguration and swarm control (this approach is also discussed by Barca and Sekercioglu [103]).

Brambilla, et al. [63] take an alternate approach in, like Abukhalil, et al. [102], categorizing prior articles on swarm intelligence. Unlike Abukhalil, Ptali, and Sobh's approach, Brambilla, et al. classify systems into the categories of method-based and collective behavior-based. The method-based category is further divided into two sub-categories (design and analysis methods), that are further divided into five sub-categories (behavior-based, automatic design, microscopic, macroscopic, and real-robot analysis). Collective behaviors are divided into four sub-categories (spatially organizing, navigation, collective decision-making, and other), which are further divided into ten sub-categories.

Swarm control has been demonstrated for a variety of applications, including robotic self-assembly [51], dynamic cleaning [104], exploration and mapping [105], foraging [106], object movement and interaction [105], and coordinating cooperation [105], [107]. Systems implementing swarm approaches, according to Sahin [101], have benefitted from system robustness, flexibility, and scalability benefits, provided by the approach. Barca and Sekercioglu [103] also identify a number of application-specific benefits [2].

3. SIMULATION[2,3]

A key challenge relating to analyzing self-replicating robot systems, and the decision-making algorithms for those systems, is that there isn't currently a standard means to simulate these systems. Thus, for the purpose of this work, a simulation system was developed to provide a means to accomplish this. In this chapter, this simulation system works is detailed.

3.1. Simulation Overview

The simulation system consists of a self-replicating robot system interacting with an environment. There is no graphical/visual aspect of the simulation system, other than textual output. The goal of the simulation system is to further understand the constraints and variables involved in self-replicating robot system configurations and decision-making strategies.

A simulation run is divided into a number of iterations (time-steps). The number of time-steps for a given run is determined by an input number. Each time-step corresponds to a block of time, with the simulation starting at step zero and progressing until the desired number of time-steps is reached. In each time-step of the simulation, robots in the robot system perform tasks which involve acquiring resources, converting resources, or assembling new robots. The following sections describe these in more detail, along with the operation of the simulation from a wholistic standpoint.

[2] This chapter is derived from: A. Jones and J. Straub, "Simulation and Analysis of Self-Replicating Robot Decision Making Systems," *Comput.*, vol. 10, no. 1, 2021.
[3] This chapter is partially derived from: A. Jones and J. Straub, "Software Simulation System for Self-Replicating Robot Decision-Making," (under preparation for submission to) SoftwareX.

3.2. Resources and Task Types

There are three different types of resources in the simulation, which were presented in previous work [2][5]. These types are as follows:

- **Nonprintable Components**: a resource type of components that the robot system doesn't have the capability to print (or make in-situ) such as control units (processors) and motors.
- **Printable Components**: a resource type of printable components that are fabricated by the robot system over the course of the simulation, such as frames (or structural elements) for new robots.
- **Raw Printing Materials**: a resource type of materials that are used in the printing process. The printing process would yield the printable component resource type, so this raw type requires a fabricating step before materials are useable to build new robots.

The simulated robot system begins the simulation run with an initial amount of each type of resource. Furthermore, the environment has a certain amount of raw printing materials available which robots could collect.

For the purposes of the simulation, each of these component types are represented by a single numerical quantity. The nonprintable components are combined into a single numerical quantity which refers to a volume (or mass) of components. In practice, the nonprintable components themselves would likely be a diverse range of different parts that wouldn't necessarily be interchangeable with one another. The simulation could be run beforehand and used to project which nonprintable components would need to be stored in that allotted volume. For the printable component resource type, the components are assumed to be interchangeable for simplicity. In practice, a system would be needed to predict what type of printable components should be crafted.

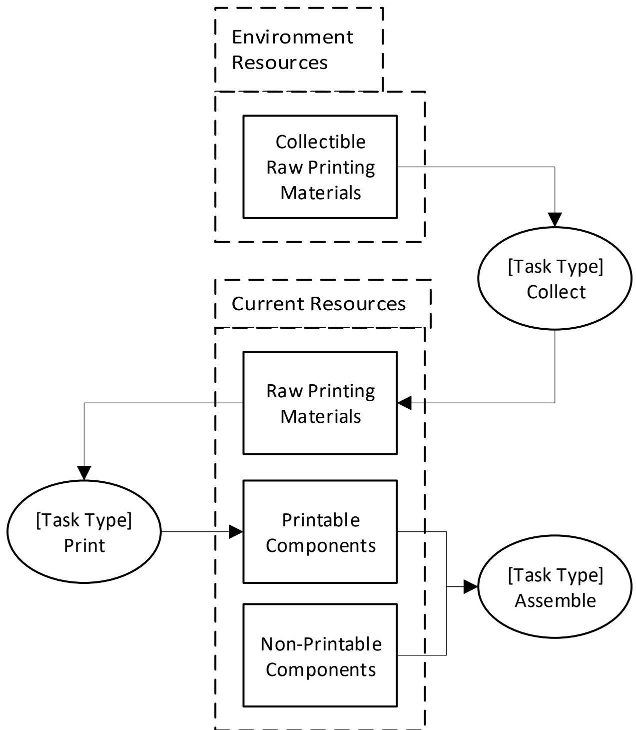


Figure 3.1. Diagram depicting how task types are related to resource types.

There are four task types in the simulation, three which perform an action (depicted in Figure 3.1) and one which represents a default state indicating that a robot is currently performing no action (idle). The description of the task types are as follows:

- **Collect**: a task type which involves a robot gathering raw printing materials from the environment and adding the gathered materials to the robot system's inventory. This task type has a time-step duration of 1, meaning that if a robot is assigned this task type it would complete it in a single time-step. Upon completion of this task, raw printing materials are removed from the environment and added to the robot system's resource pool. The amount collected upon completion is a parameter (*Collect_Amount*), which represents an average amount returned per time-step (as it would vary based on resource sparsity).

- **Print**: a task type which involves a robot taking raw printing materials and crafting them into printable components. This task type has a time-step duration of 1, meaning that if a robot is assigned this task type it would complete it in a single time-step. Upon completion of this task, raw printing materials would be removed (*Print_Amount*) from the robot system's resource pool and printable components would be added to the robot system's resource pool. This represents an average amount of components printed per time-step (including failed prints). The conversion factor of raw printing materials to printable components is a parameter (*Print_Efficiency*).
- **Assemble**: a task type which involves a robot taking nonprintable components and printable components from the robot system's resource pool and assembling them into a new robot. This task type has a duration that varies by the robot type being assembled. Upon completion of this task, the newly assembled robot would be added to the robot system.
- **Idle**: a default task type which is assigned to any robot not performing any other action during a time-step. This task type has no associated duration, because it doesn't have any completion actions/events.

3.3. Robot Types

In the simulation, there are four types of robots: normal, printer, assembler, and replicator. In each time-step, each robot is either idle, gathering resources, printing components, or assembling a new robot. However, certain robot types are restricted in what types of tasks that they can perform (listed in Table 3.1). All robot types are capable of being idle. Not all types of robots must be included in any given simulation run.

The robots in the simulation are single-task robots that carry out single-robot tasks with a time-extended assignment. Thus, the robot systems in the simulation would be categorized as ST-SR-TA, according to the task allocation taxonomy proposed by Gerkey in [108].

Table 3.1. Robot capabilities based on type of robot.

Robot Type	***Collect Resources***	***Print Components***	***Assemble Robots***
Normal	●		
Assembler	●		●
Printer	●	●	
Replicator	●	●	●

The material cost of each robot type is directly related to its capabilities. Capability costs for each included capability are added together to determine the cost of the robot type. For example, the normal robot type cost is just the base cost, while the printer robot type's cost is calculated by adding the base cost plus the printing capability cost. The default values for these costs are listed in Table 3.2. These values are simulation parameters and can be adjusted from run to run of the simulation, to determine what is being studied by each simulation run.

Table 3.2. Default resource costs by capability.

Cost per Capability	**Nonprintable Cost**	**Printable Cost**	**Build Duration Cost**
Base (Collect Capability)	1	2	2
Printing Capability	1	2	2
Assembly Capability	1	2	2

3.4. Replication System Configurations

In this section, the system configurations of self-replicating robot systems used in the simulation are discussed and presented. The replication system configurations are a combination of choices from two sets. The first set, the replication approach, consists of centralized, decentralized, and hierarchical. This determines how many robots have a replication-related capability. The replication-related capabilities, for this simulation, include the assembly and print task types. The characteristics of this set are as follows:

- **Centralized**: robots that have a replication-related capability are not buildable by the robot system. These systems exclusively use initial/factory-made robots for replication related capabilities.
- **Decentralized**: all robots in this type of system have one or more replication-related capabilities. These systems require that built robots have some capability in terms of replication.
- **Hierarchical**: buildable robots in this type of system may or may not have replication-related capabilities. These systems are a combination of the centralized and decentralized approaches and do not impose strict replication-related capability requirements on buildable robot types (although at least one of each is present).

The second set, the production approach, consists of homogeneous and heterogeneous. This set is combined with the previous set to derive the replication system configurations used for the simulation (listed in Table 3.3). The characteristics of the members of this set are as follows:

- **Homogeneous**: these systems have a single robot type for all the replication-related capabilities. In the listed simulation robot types, this is the replicator robot type.

- **Heterogeneous**: these systems have multiple robot types that have replication-related capabilities. In the listed simulation robot types, this consists of the assembler and printer robot types.

Table 3.3. Buildable robot types by system configuration.

Buildable Robot Types	Centralized	Decentralized	Hierarchical
Homogeneous	Normal	Replicator	Replicator, Normal,
Heterogeneous	Normal	Assembler, Printer	Assembler, Printer, Normal

Based on the combination of the selections from the two sets, the utilized replication system configurations are formed. The system configurations of self-replicating robot systems used in this work are listed in Table 3.4. The variable ‘n’ denotes the total number of robots in the system. Thus, as the number of robots in the system changes, the ratio of the robot types is related to ‘n’. Any range that contains ‘n’ indicates that that robot type is buildable in that system configuration.

Table 3.4. System configurations utilized in the simulation system.

System Configuration	*Robot Type*			
	Normal	Printers	Assemblers	Replicators
Centralized - Homogeneous	n - 1	-	-	1
Decentralized - Homogeneous	-	-	-	n
Hierarchical - Homogeneous	[1, n - 1]	-	-	[1, n - 1]
Centralized - Heterogeneous	n - 2	1	1	-
Decentralized - Heterogeneous	-	[1, n - 1]	[1, n - 1]	-
Hierarchical - Heterogeneous	[1, n - 2]	[1, n - 2]	[1, n - 2]	-

The details of each of the system configurations are as follows:

- **Centralized Homogeneous (CHO)**: One robot is responsible for both printing components and assembling them. Constructed robots are of the normal type and either gather resources or complete other objectives.
- **Decentralized Homogeneous (DHO)**: All robots have the capability to print components, assemble them, and gather resources or complete other objectives.

- **Hierarchical Homogeneous (HHO)**: There are a variable number of robots capable of printing components and assembling them. There are also a variable number of normal type robots.
- **Centralized Heterogeneous (CHE)**: One robot is responsible for printing components, and another (distinct) robot is responsible for assembling them. Constructed robots are of the normal type and either gather resources or complete other objectives.
- **Decentralized Heterogeneous (DHE)**: Robots have either the capability to print components or the capability to assemble them. All robots can gather resources or complete other objectives.
- **Hierarchical Heterogeneous (HHE):** There are a variable number of robots capable of printing components, a variable number capable of assembling them (distinct from printing group), and a variable number of normal type robots. All robots can gather resources or complete other objectives.

For the purposes of the simulation, the homogeneous systems start with a single replicator robot, while the heterogeneous systems start with two robots - an assembler robot and a printer robot. The non-assembly capable robot types would have a maximum of 'n' – 1 in all of the configurations since, without a robot capable of assembling additional robots, the robot system would not be considered to be self-replicating by any version of the definition.

3.5. Robot Build Quality and Task Risks

In this section, the topics of robot build quality and the risks of each task type are discussed.

3.5.1. Robot Build Quality

An important capability of a self-replicating robot system is the ability to fabricate parts and assemble new robots. This introduces the question of the quality of the built robot, as a robot built in-situ may have quality defects (without the ability to simply discard it with minimal impact, like in a factory setting). To facilitate assessment, the simulation assigns each robot a build quality. A robot's build quality value ranges from zero to one, with one being very high quality and zero being very poor quality. The quality value is a decimal value and not a binary one or zero value.

Parameters are used with estimated values to model possible changes in build quality. These parameters and their default values are listed in Table 3.5. These values are rough predictions used to demonstrate principles. The actual quality metrics would vary significantly, based on hardware aspects and operating environment. Furthermore, these values are varied during experimentation in order to determine their relative impact on the system. In this regard, the relative impact of build quality, from generation to generation, is of interest (more than the precise values).

Table 3.5. Parameters for assigning the build quality of newly assembled robots.

Build Quality Determination	*Chance*	*Lower Bound*	*Upper Bound*
Quality Increase *(Quality_incr)*	5%	0.01	0.05
Quality Decrease *(Quality_decr)*	50%	0.01	0.25

The build quality value is determined as follows. When a new robot is built, it has a build quality value determined based on the robot that assembled it. First, a random value is generated, and if the value is above one minus the increase quality chance (the default value for this is listed in Table 3.5), then the newly assembled robot's build quality equals the build quality of the assembling robot plus a random number between the increase quality lower and upper bound parameter values (zero and one are a strictly enforced minimum and maximum). If not within the increase chance, then it is compared to the decrease quality chance parameter. Similar to before, if the random number is below the decrease quality chance then the newly assembled robot's build quality equals its builder's quality minus a random number between the upper and lower bounds. The pseudocode for this process is shown in Equation 2. The variable $Robot_{Quality}$ refers to the build quality of the newly constructed robot, $Assembler_{Quality}$ refers to the build quality of the robot that assembled it, and the quality parameters refer to the values in Table 3.5.

```
rand ← random(0, 1)
if rand > (1.0 - Quality_incr_Chance):
   Robot_Quality ← Assembler_Quality + random(Quality_incr_Lower, Quality_incr_Upper)
else if rand < Quality_decr_Chance:
   Robot_Quality ← Assembler_Quality - random(Quality_decr_Lower, Quality_decr_Upper)
else:
   Robot_Quality ← Assembler_Quality
```

(Eq. 2)

In the simulation, a robot's build quality has three effects. First and foremost, there is a basic functionality quality threshold parameter (*QualityThreshold*). If the quality of a newly built robot is below the quality threshold parameter, then the robot is considered non-functional and destroyed in the simulation. The rationale behind this is that a robot of sufficiently low quality would fall apart or be otherwise non-functional for the purposes of task assignment. A factor that may heavily contribute to this is the robot system's operating environment. More hazardous environments, for example, may be less forgiving in terms of defects. The second effect of build quality is that the quality of the assembling robot is used to determine the quality of the built robot. Thus, this effect only impacts assembly-capable robot types. The rationale behind this is that a robot with defects may not be capable of assembling a robot as well as one without (or with less) defects. Third, a robot's build quality is a factor in task risk calculations, as described below.

3.5.2. Task Risks

Tasks have a risk of failing in a manner which harms or destroys the robot performing the task. For the purpose of the simulation, this is taken into account with the parameters listed in Table 3.6. The listed risk amount values are adjustable parameters. The rationale behind the particular consequence of each task type is as follows. Collecting resources involves robots foraging in an environment. This puts them at risk of being stuck, trapped, navigationally lost, or otherwise lost due to environmental hazards. This isn't the case with printing or assembling, as navigating the environment wouldn't be part of these tasks. The task of printing was chosen to have the consequence of the robot's printing capabilities being lost due to a significant hardware malfunction occurring. Similarly, the assemble task has the potential consequence of the assembly hardware breaking down. The default risk amounts were chosen based on estimations but would likely vary significantly based on hardware specifications and environmental factors. These are configurable parameters within the simulation system, to facilitate experimentation.

Table 3.6. Default values for risk chance and consequences of tasks based on task type.

Task Type:	*Collect*	*Print*	*Assemble*
$\boldsymbol{Risk_{Task\_Type}}$	1%	0.1%	0.1%
Consequence	Robot Lost	Printing Capability Lost	Assembly Capability Lost

The build quality of the robot performing a task may also affect the associated risks of performing a given task. The rationale behind this is that as the quality of a robot decreases, the risk of it encountering problems performing a task would likely increase. Thus, Equation 3 was utilized in order to address this. The variable $Risk_{Task\_Type}$ refers to the values in Table 3.6 (corresponding to the task type) and the variable $Robot_{Quality}$ refers to the build quality of the robot performing the task.

$$Risk_{Task} = Risk_{Task\_Type} + \left(1.0 - Robot_{Quality}\right) * Risk_{Task\_Type} * RiskQuality\_Modifier$$
$$if\ Robot \in FactoryMade\text{:}\ \ Risk_{Task} = Risk_{Task} * RiskFactory\_Modifier \qquad \text{(Eq. 3)}$$

In Equation 3, a robot's lack of quality increases the risk in a manner related to the task type's risk. Omitting $Risk_{Task\_Type}$ in the multiplication would increase the risk by a constant, which could end up being a dominating factor and negating the relevance of the task's inherit risk. A risk modifier parameter (*RiskQuality_Modifier*) is then applied in order to add a weighting factor for the impact of quality on the risk.

Another consideration is that the initial robots of the robot system would likely be factory built and tested to a high degree. Thus, the associated risk amount of these (likely thoroughly tested) initial factory-built robots would be less than robots built by the robot system. In this regard, a parameter (*RiskFactory_Modifier*) is used to scale the risk cost of factory-built robots performing a task.

3.6. Simulation Operation

In this section, the programmatic flow and operation of the simulation is detailed. The simulation is divided into a number of iterations (time-steps), determined by an input parameter. Each time-step corresponds to a block of time, with the simulation starting at step zero and progressing until the input number of time-steps is reached. A time-step corresponds to an iteration of the main loop in Figure 3.2. In the main loop, the simulation events are evaluated followed by decision-making (planning) by the robot system. These are discussed in the following subsections.

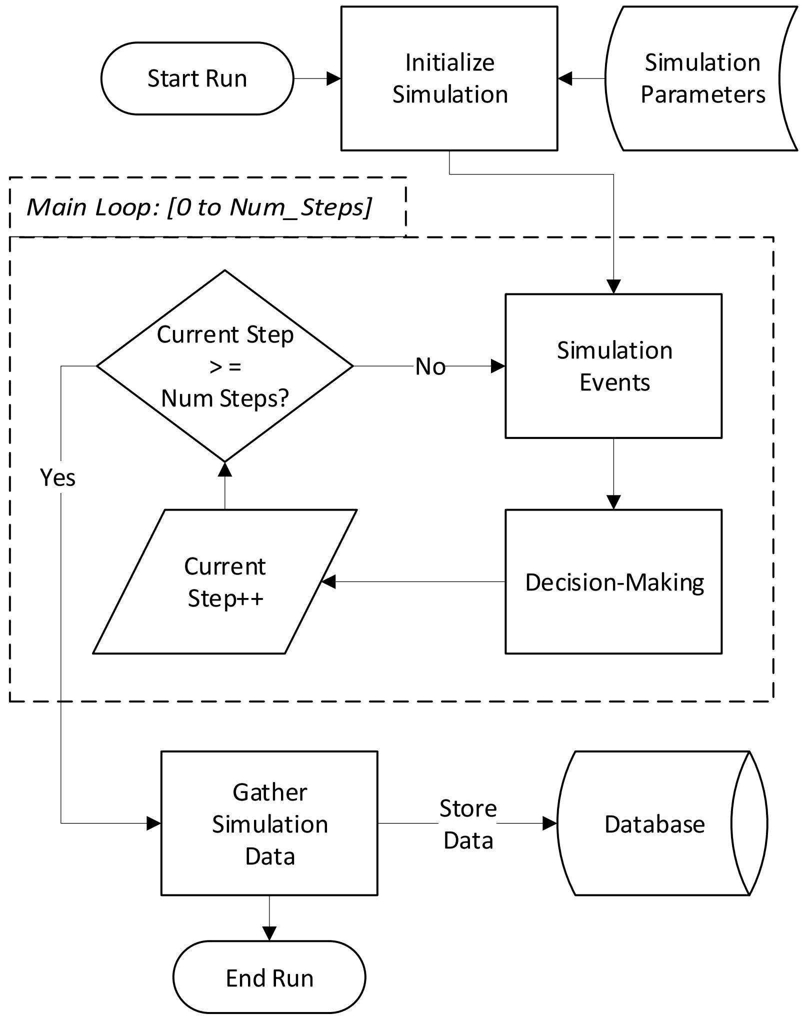


Figure 3.2. Diagram of the high-level operation of the simulation.

3.6.1. Simulation Events

The simulation events process informs robots of the results of the current actions of the robot system, while the decision-making process is used for determining what actions to perform in future steps. In each time-step of the simulation, robots in the robot system perform tasks which involve acquiring resources, converting resources, or assembling new robots. The simulation events process determines the results of these actions by the robot system.

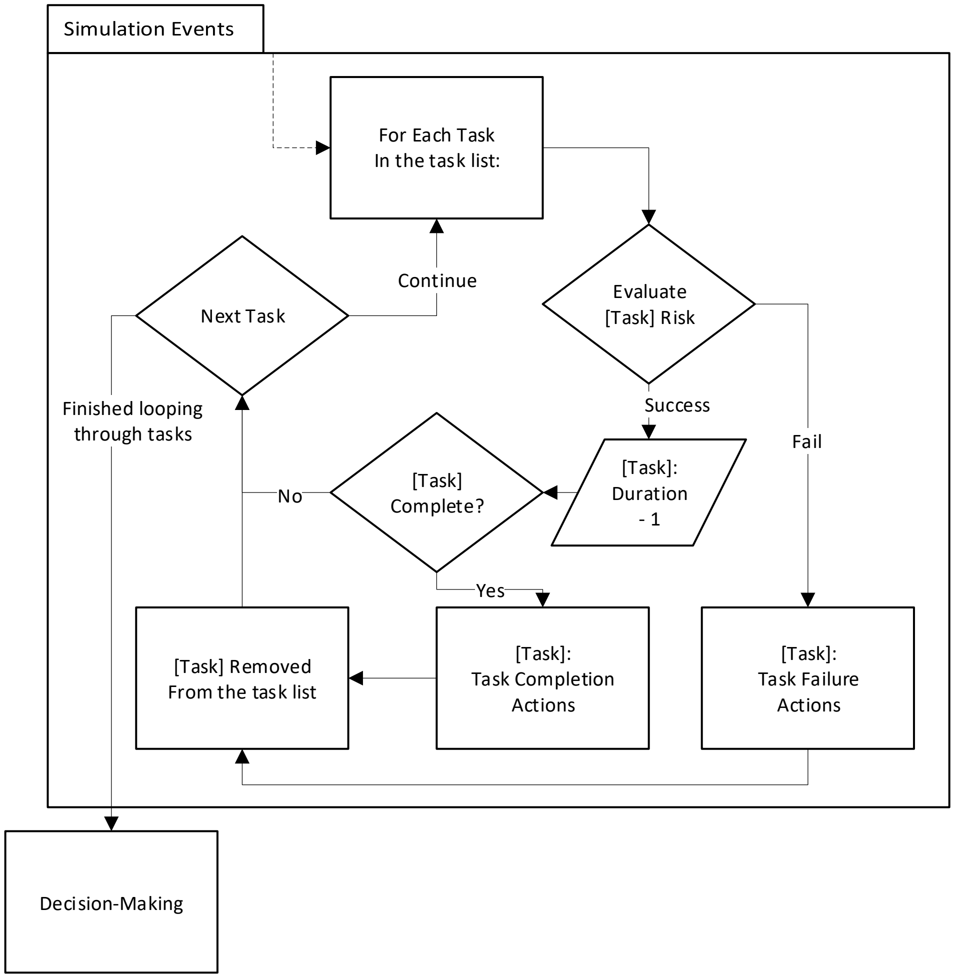


Figure 3.3. Diagram depicting the operation of simulation events.

The process for processing simulation events, depicted in Figure 3.3, consists of looping through the tasks currently being performed. For each of the tasks in progress by the robot system, the risk of performing the task is used to determine (with random input) if it failed or succeeded in the current time-step. If it succeeded, the task's remaining duration is decremented by one. If its remaining duration is now zero, then its completion actions are performed (i.e., the resources are gathered, the part is fabricated, or the robot is assembled). In the case where it failed, then the task failure actions are performed, and it is removed from the active tasks, as a failed task is abandoned.

3.6.2. Decision-Making

The decision-making process primarily consists of the robot system assigning tasks to the robots. The decision-making algorithm is setup so that it functions as a programmatic interface (i.e., a different algorithm could instantiate the interface and be used instead of the base algorithm). In this sense, it takes the robot system as an input and assigns the tasks to the robots in the system (depicted in Figure 3.4).

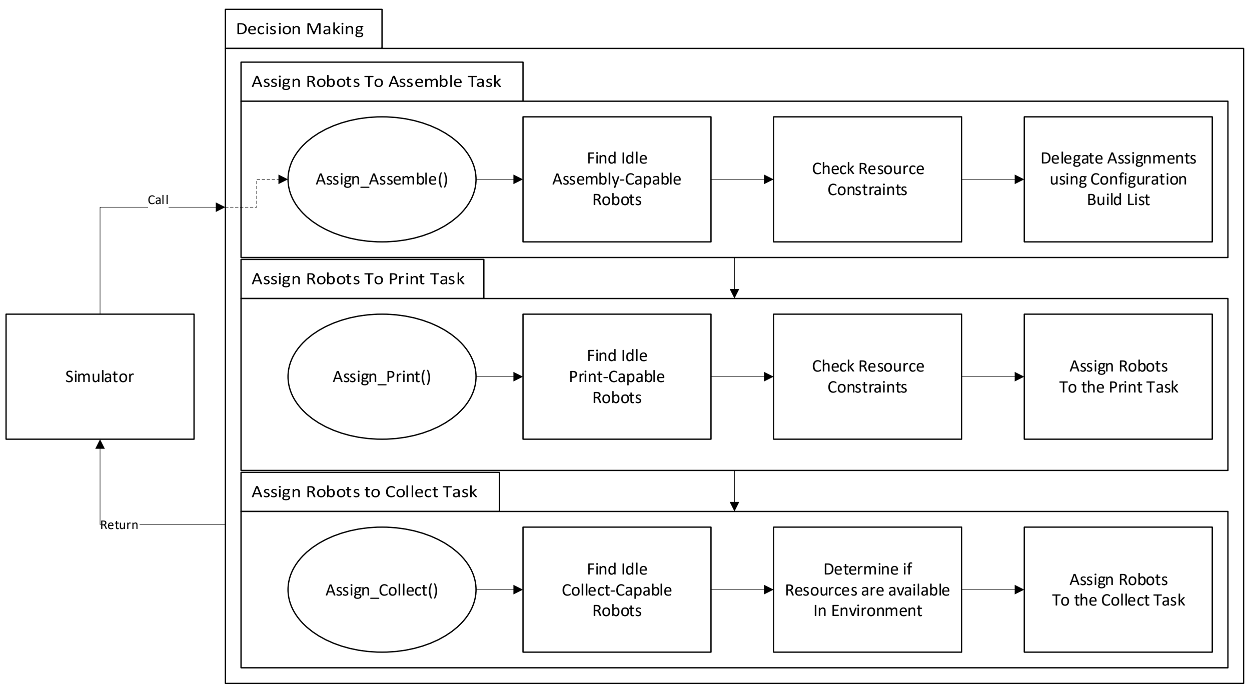


Figure 3.4. Overview of the decision-making algorithm's task assignments to the robot system.

Certain parameters and variables in the simulation are considered known to the robot system, while other parameters are considered unknown. The unknown parameters aren't available to the decision-making algorithm. These include task risk and robot build quality related values. The decision-making system could predict these values over time, based on observations. However, the exact values used are unavailable to it. Examples of known parameters include the build costs of each robot type, the robot system's current robots (including characteristics of each robot), and the robot system's current resources.

The details of the utilized decision-making algorithms are presented in subsequent chapters. The next chapter discusses using a base decision-making algorithm and comparing the results from the simulation runs in order to derive decision-making criteria. The determined decision-making criteria are then used in a subsequent chapter to derive and compare different decision-making algorithms.

3.6.3. Stochastic Processes in the Simulation System

The simulation system has certain procedures and parameters that are stochastic. Due to this, the experiments conducted are run one hundred times (with the same conditions/inputs) and have the results averaged together from those runs.

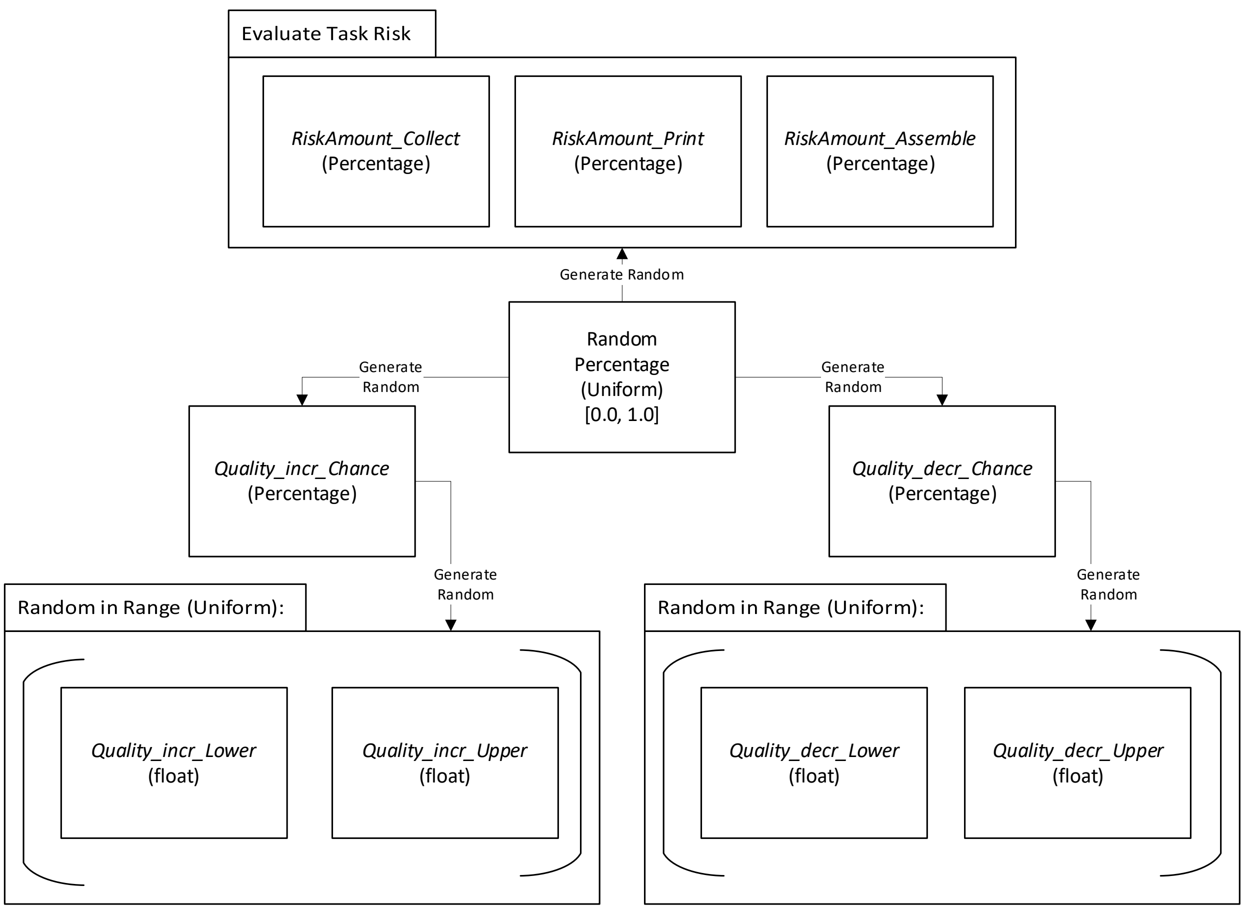


Figure 3.5. Diagram of the role of stochastic processes in the simulation system.

The components of the simulation system that involve stochastic processes include: the determination of build quality for newly assembled robots and the evaluation of task risk (depicted in Figure 3.5). The equation for determining build quality was provided in Equation 2, and the equation for evaluating task risk was provided in Equation 3. Averaging the results over one hundred simulation runs is used to account for these stochastic aspects of the simulation system in the experiments conducted in subsequent chapters.

3.6.4. Simulation Parameters

Simulation parameters are used as inputs into each simulation run. These values are varied between the experimental conditions (introduced in subsequent chapters) in order to facilitate analysis of their impact on the outcome of each simulation run. The simulation parameters and their default values (when not being altered for a particular experimental condition) are listed and described in Tables 3.7 and 3.8.

Table 3.7. Description of simulation parameters (part 1).

Parameter	**Default Value**	**Description**
Num_Steps	-	Number of iterations/time-steps that the simulation goes through.
Num_Runs	100	Number of times the simulation is run with the input parameters.
Initial_NonPr	300.0	The robot system's starting quantity of nonprintable components. This parameter is an upper bound on the robot system's growth [2]. It is initially set to a value 3x that of the initial printable components.
Initial_Printable	100.0	The robot system's starting quantity of printable components. This would vary based on mission-related constraints [2].
Initial_Materials	50.0	The robot system's starting quantity of raw printing materials. This would vary based on mission-related constraints. Initially set to ½ that of the initial printable components.
Env_Materials	500.0	The environment's quantity of collectable raw printing materials. Represents the amount of in-situ resources that can be utilized for the robot system [12]. Initially set to greater than the initial nonprintable in order to simulate an environment with abundant (but not limitless) in-situ resource availability.
BaseCost_NonPr	1	Base robot cost of nonprintable components. Initially set to 1, with equal volume of nonprintable components needed for all capabilities.
PrintCost_NonPr	1	Print capability cost of nonprintable components. Initially set to 1, with equal volume of nonprintable components needed for all capabilities.
AssembleCost_NonPr	1	Assemble capability cost of nonprintable components. Initially set to 1, with equal volume of nonprintable components needed for all capabilities.
BaseCost_Pr	2	Base robot cost of printable components. Initially set to 2 to .
PrintCost_Pr	2	Print capability cost of printable components. Initially set to 2x the cost of print-capability nonprintable components.
AssembleCost_Pr	2	Assemble capability cost of printable components. Initially set to 2x the cost of assemble-capability nonprintable components.
BaseCost_Time	2	Base robot cost of build time. This simulates the amount of time it takes to assemble a base robot. Initially set to 2 based on ratio of collection and printing of components.
PrintCost_Time	2	Print capability cost of build time. This simulates the amount of additional time needed to add the print capability to a robot under construction. Initially set to be equal to *BaseCost_Time* in order for the time increase to be linear.
AssembleCost_Time	2	Assemble capability cost of build time. This simulates the amount of additional time needed to add the assemble capability to a robot under construction. Initially set to be equal to *BaseCost_Time* in order for the time increase to be linear.
Print_Efficiency	1.0	Factor that scales raw printing materials to printable components. Conversion ratio from raw material quantities to printable component quantities. Initially set to 1:1 conversion.
Print_Amount	1.0	Amount of raw materials converted per print task. Quantity that is scaled by the print efficiency parameter to determine how many raw materials are converted per print task completion.
Collect_Amount	1.0	Raw printing materials per collecting robot per timestep.

Table 3.8. Description of simulation parameters (part 2).

Parameter	**Default Value**	**Description**
QualityThreshold	0.5	Robots with a quality below this are non-functional. This parameter is used to model the build quality at which the necessary functionality of the robot is degraded beyond usability. Initially set to 0.5 such that the quality of generations has a mid-level threshold.
Quality_incr_Chance	5.0%	Chance that a new robot's build quality will increase. This value was chosen based on the comparatively low estimated percentage chance of the build quality increasing over generations for self-replicating robot systems, although this would have hardware dependencies. This is based on previous work in [5]. Initial value of 1/10th to that of the decrease chance.
Quality_incr_Lower	0.01	Lower bound for quality increase amount. Initially set to 0.01 such that the range does not have a significant minimum quality increase.
Quality_incr_Upper	0.05	Upper bound for quality increase amount. This value is less than that of the upper bound of the decrease chance due to a lower quality assembling robot potentially not being able to assemble a robot with a quality much greater than its own [5]. Although this is impacted by hardware implementation.
Quality_decr_Chance	50.0%	Chance that a new robot's build quality will decrease. This value was chosen based on the comparatively high estimated percentage chance of the build quality decreasing over generations for self-replicating robot systems, although this would have hardware dependencies. This is based on previous work in [5]. Initial ratio of 10x that of the increase chance.
Quality_decr_Lower	0.01	Lower bound for quality decrease amount. Initially set to 0.01 such that the range does not have a significant minimum quality decrease.
Quality_decr_Upper	0.25	Upper bound for quality decrease amount. This value is greater than that of the upper bound of the increase chance due to both the predicted probability and severity of defects occurring over generations [5]. Although this is impacted by hardware implementation.
RiskAmount_Collect	1.0%	Risk chance for the collect task type. This signifies the estimated foraging risk, which is impacted by environment and robotic hardware aspects [2]. Initially set to 1% to simulate a moderate foraging risk.
RiskAmount_Assemble	0.1%	Risk chance for the assemble task type. Assembly equipment break-down rate initially set to 1/1000th , although this would vary based on hardware and the operating environment (among other factors).
RiskAmount_Print	0.1%	Risk chance for the print task type. 3D printer break-down rate initially set to 1/1000th , although this would vary based on hardware and the operating environment (among other factors).
RiskQuality_Modifier	5.0	Multiplier for impact of quality defects on risk amount. The initial value of 5 was chosen based on trials of the simulation system.
RiskFactory_Modifier	0.1	Multiplier for impact of factory-made robots on risk amount. Thoroughly tested robots would have a lower probability of failure. This initial value would scale the risk by 1/10th.

3.6.5. Software Overview

The implementation of the simulation system used in the subsequent chapters was written in the Python programming language. An overview class diagram of the program is depicted in Figure 3.6.

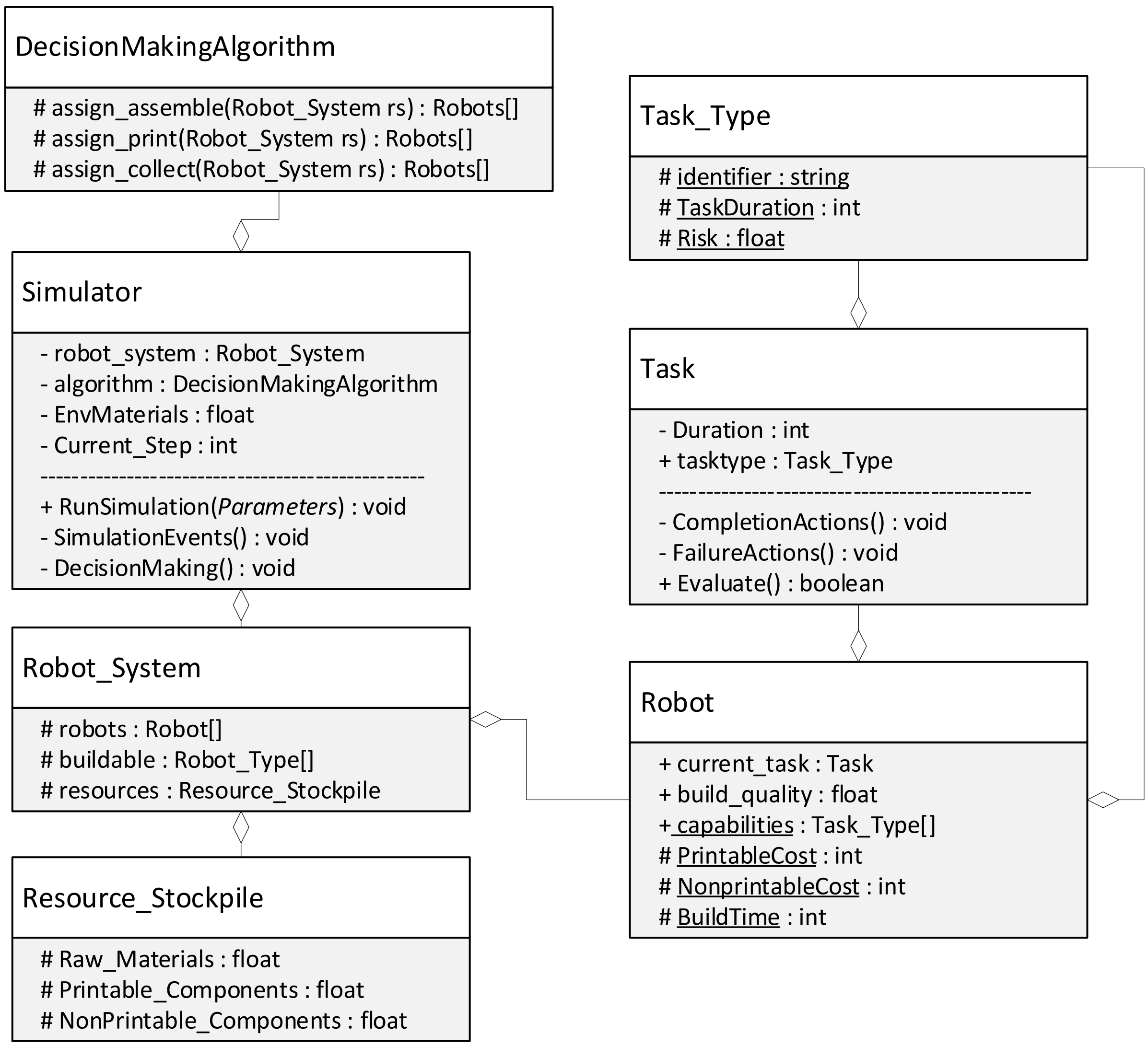


Figure 3.6. Overview class diagram of the utilized simulation system implementation.

The design of the simulation system was focused on the use of encapsulation in order to allow changes and new features to be integrated more easily. The development of the software went through three separate iterations: base functionality, experimental condition features (discussed in Chapter 4), and cybersecurity features (discussed in Chapter 6, not shown in Figure 3.6). 'Bug fixing' and validation of previous features was conducted to ensure functionality during each of the three development iterations. The functionality of the simulation system was verified through several test runs and the step by step logging of events and variables.

The 'Simulator' class is responsible for running the simulation. It is first initialized with an instance of the 'DecisionMakingAlgorithm' class. Then, the input parameters are passed to it via the 'RunSimulation' method, which runs the simulation *Num_Runs* amount of times. The results of each run are stored in memory and the average of each result field is taken once all the runs have completed. The averaged results are written to a CSV file (which is subsequently imported and merged with the data in the database).

The 'Robot' class is the base class for each robot type used in the simulation. A sub-class is made for each robot type, which modifies the capabilities variable based on the task types that it is capable of performing. The costs are based on parameters and are set by the 'Simulator' class at the start of each run. The 'Task_Type' class is also a base class for each task type used in the simulation, where the duration and risk of each sub-class is changed by the 'Simulator' based on parameter values. Similarly, the 'DecisionMakingAlgorithm' class is also the base class which each of the decision-making algorithms (discussed in the subsequent chapters) inherit.

4. SYSTEM CONFIGURATION EXPERIMENT[4]

In this chapter, an experiment that was performed utilizing the simulation system described in the previous chapter is presented and analyzed.

4.1. Experiment Overview

The goal of this experiment was to collect data and use the knowledge gained from its analysis to develop decision-making criteria for a decision-making algorithm. The decision-making algorithms are derived and compared in Chapter 5. This chapter focuses on establishing the decision-making criteria for the decision-making algorithms to utilize. In this section, the system configurations that were utilized are reviewed, and the base decision algorithm used for the experiment is presented.

4.1.1. Replication System Configurations

In this subsection, information on the replication system configuration (discussed in the previous chapter) is provided for reference. Two sets of approaches are combined to form the replication system configurations used in the experiment. The result of combining the higher-level categories is listed in Table 4.1. The first set (replication approach) is as follows:

- **Centralized (prefix: C)**: A system where robots that have a replication-related capability are not buildable by the robot system. These systems exclusively use pre-existing, factory-made robots to provide replication related capabilities.

---

[4] This chapter is derived from: A. Jones and J. Straub, "Simulation and Analysis of Self-Replicating Robot Decision Making Systems," *Comput.*, vol. 10, no. 1, 2021.

- **Decentralized (prefix: D)**: A system where all robots have one or more replication-related capabilities. These systems mandate that all built robots have some capability in terms of replication.
- **Hierarchical (prefix: H)**: A system where buildable robots may or may not have replication-related capabilities. These systems are a combination of the centralized and decentralized approaches and do not impose strict replication-related capability requirements on buildable robot types (although at least one of each is present).

The first set is then combined with the second set of categories. The second set (production approach) is as follows.

- **Homogeneous (suffix: HO)**: a system that uses a single robot type for all the replication-related capabilities. In the current simulation setup, this is the replicator robot type.
- **Heterogeneous (suffix: HE)**: a system that uses multiple robot types that have replication-related capabilities. In the current simulation setup, this consists of the assembler and printer robot types.

Table 4.1. Buildable robot types by system configuration.

System Configuration		**Robot Type**			
ID	**Name**	**Normal**	**Printers**	**Assemblers**	**Replicators**
CHO	Centralized Homogeneous	●	-	-	○
DHO	Decentralized Homogeneous	-	-	-	●
HHO	Hierarchical Homogeneous	●	-	-	●
CHE	Centralized Heterogeneous	●	○	○	-
DHE	Decentralized Heterogeneous	-	●	●	-
HHE	Hierarchical Heterogeneous	●	●	●	-

*Buildable robot types are denoted with the filled in circle, and robots that are present (but not buildable) are denoted with a hollow circle.

4.1.2. Base Decision-Making Algorithm

To perform the experiment in this chapter, a simplistic decision-making algorithm was utilized to analyze trends in the performance of the different system configurations under multiple experimental conditions. This decision-making algorithm is now discussed.

In the base decision-making algorithm, the choice of when to build a new robot, and what type it should be, is decided with simple criteria. This involves finding all idle assembly-capable robots and then taking the buildable robot type list (see Table 4.2) and repeating it (in order) until there are enough robots in the build queue for each idle assembly-capable robot. Resource constraints are not checked initially. If a robot cannot be built, the would-be assembler simply remains idle. The analysis and the improvement of this portion of the decision-making system is a key goal of this experiment. The impact and consideration of multiple factors may allow an improved decision-making system to make more optimal choices based on the criteria considered.

Table 4.2. Robot build list by system configuration.

Buildable Robot Types	**Centralized**	**Decentralized**	**Hierarchical**
Homogeneous	Normal	Replicator	Replicator, Normal
Heterogeneous	Normal	Assembler, Printer	Assembler, Printer, Normal

After determining what each of the robots need to assemble (if anything), the base decision-making algorithm assigns all currently idle print-capable robots to fabricate printable components. This is limited by the robot system's current amount of available raw printing materials (robots will not be assigned to printing tasks that materials are not available for). This is shown in Equation 4, where 'AssignedPrint' represents the set of robots that are assigned the print task at a given time-step and the 'MaxPrint' variable denotes the maximum number or robots that could print given the robot systems' current supply of raw materials.

$$
\begin{aligned}
&MaxPrint := \left\lfloor \frac{RobotSystem_{RawMaterials}}{Print_{Amount}} \right\rfloor \\
&CanPrint := \{\forall Robot: (Robot_{CurrentTask} == Idle) \land (Print \in Robot_{Capabilities})\} \\
&AssignedPrint := S \mid (S \subseteq CanPrint) \land (|S| == MIN(|CanPrint|, MaxPrint)) \\
&\ldots \\
&AssignedCollect := \{\forall Robot: (Robot_{CurrentTask} == Idle) \land (Collect \in Robot_{Capabilities})\}
\end{aligned}
\qquad \text{(Eq. 4)}
$$

After these assignments, all robots that are idle are assigned to collect materials from the environment. This is represented by the 'AssignedCollect' set of robots in Equation 4. If robots return no materials, then it is assumed that the environment is out of raw materials and the system stops assigning any robots to the collection task once this happens. The task risk associated with the collection task is the motivation for discontinuing unfruitful collection. However, the stage at which it can be assumed that no further resources are available to collect may be more complex in real world instances.

4.2. Experiment Methodology

In this section, the metrics, experimental conditions, and hypotheses are presented.

4.2.1. Output Metrics

There are two types of metrics recorded for each simulation run. These types are primary and secondary. The primary metrics are categorized as such since they are of higher interest than the secondary metrics for the purposes of this experiment. The primary metrics measured and recorded by the simulation include:

- **Assembly Potential:** the number of robots that have the assembly capability at the end of a simulation run. This includes replicator and assembler robot types, which haven't succumbed to a task risk and lost their capability. The standard deviation of this value, among the simulation runs, is also recorded.

- **Collection potential:** the number of robots that have the collect capability at the end of a simulation run. All robot types have this capability in this simulation, so this is always equal to the current number of robots in the system. The standard deviation of this value, among the simulation runs, is also recorded.
- **Print Potential:** the number of robots that have the print capability at the end of a simulation run. This includes replicator and printer robot types, which haven't succumbed to a task risk and lost their capability. The standard deviation of this value, among the simulation runs, is also recorded.

The secondary metrics measured and recorded by the simulation include:

- **Current Number of Robots:** The current number of robots in the system during the final time-step of the simulation. This does not include robots that were destroyed or lost due to succumbing to task risks.
- **Total Number of Robots:** The total number of robots in the system. This includes robots that were destroyed or lost due to task risks.
- **Average Robot Quality:** The average build quality of the current robots in the system during the final time-step of the simulation.
- **Assemble Ratio:** The ratio of robots that have the assemble capability to the current overall number of robots.
- **Print Ratio:** The ratio of robots that have the print capability to the current overall number of robots.
- **Collect Ratio:** The ratio of robots that have the collect capability to the current overall number of robots. In the current simulation setup, this is the same as the current number of robots since they all have the collect capability.

- **Number of Robots Destroyed (Build Quality):** The number of robots that, when assembled, had a quality lower than the *QualityThreshold* parameter (and were therefore considered non-functional at the end of their fabrication process).
- **Number of Robots Destroyed (Task Risk):** The number of robots that were destroyed due to succumbing to task risks. For the current simulation setup, this is a hazard risk of the collect task type.
- **Number of Capabilities Lost:** The number of robots that lost one or more capabilities due to succumbing to a task risk. For the current simulation setup, this is a hazard of the print and assemble task types.

4.2.2. Simulation Parameters

Simulation parameters are used as inputs into each simulation run. These values are varied between the experimental conditions in order to facilitate analysis of their impact on the outcome of each simulation run. The simulation parameters and their default values (when not being altered for a particular experimental condition) are listed in Table 4.3.

Table 4.3. List and description of the simulation parameters.

Parameter	Default Value	Description
Num_Steps	-	Number of iterations/time-steps that the simulation goes through.
Num_Runs	100	Number of times the simulation is run with the input parameters.
Initial_NonPr	300.0	The robot system's starting quantity of nonprintable components.
Initial_Printable	100.0	The robot system's starting quantity of printable components.
Initial_Materials	50.0	The robot system's starting quantity of raw printing materials.
Env_Materials	500.0	The environment's quantity of collectable raw printing materials.
BaseCost_NonPr	1	Base robot cost of nonprintable components.
PrintCost_NonPr	1	Print capability cost of nonprintable components.
AssembleCost_NonPr	1	Assemble capability cost of nonprintable components.
BaseCost_Pr	2	Base robot cost of printable components.
PrintCost_Pr	2	Print capability cost of printable components.
AssembleCost_Pr	2	Assemble capability cost of printable components.
BaseCost_Time	2	Base robot cost of build time.
PrintCost_Time	2	Print capability cost of build time
AssembleCost_Time	2	Assemble capability cost of build time.
Print_Efficiency	1.0	Factor that scales raw printing materials to printable components.
Print_Amount	1.0	Amount of raw materials converted per print task.
Collect_Amount	1.0	Raw printing materials per collecting robot per timestep.
QualityThreshold	0.5	Robots with a quality below this are non-functional.
Quality_incr_Chance	5.0%	Chance that a new robot's build quality will increase.
Quality_incr_Lower	0.01	Lower bound for quality increase amount.
Quality_incr_Upper	0.05	Upper bound for quality increase amount.
Quality_decr_Chance	50.0%	Chance that a new robot's build quality will decrease.
Quality_decr_Lower	0.01	Lower bound for quality decrease amount.
Quality_decr_Upper	0.25	Upper bound for quality decrease amount.
RiskAmount_Collect	1.0%	Risk chance for the collect task type.
RiskAmount_Assemble	0.1%	Risk chance for the assemble task type.
RiskAmount_Print	0.1%	Risk chance for the print task type.
RiskQuality_Modifier	5.0	Multiplier for impact of quality defects on risk amount.
RiskFactory_Modifier	0.1	Multiplier for impact of factory-made robots on risk amount.

4.2.3. Experimental Conditions

Each experimental condition was run with three different time-step inputs ("*Num_Steps*" parameter). These input values were 30, 50, and 70. As listed in Table 4.4, each experimental condition was run one hundred times for each system configuration and time-step input. The results were averaged together from the runs of each system configuration on each experimental condition and time-step.

Table 4.4. Breakdown of number of runs per experimental condition.

Number of time-steps (*Num_Steps*)	**30 Steps**	**50 Steps**	**70 Steps**	
Number of runs per configuration (*Num_Runs*)	100	100	100	*Total*
Runs per experimental condition	600	600	600	1800

The experimental conditions are divided into four classifications. Members of experimental classification ‘A’, which are listed in Table 4.5, involve varying the build costs for the different robot types. These costs include the printable and nonprintable components required, as well as the build duration. Second, experimental condition classification ‘B’ (listed in Table 4.6) involves varying the parameters: *Print_Efficiency*, *Print_Amount*, and *Collect_Amount*. These parameters affect the resource gathering and conversion ratios. Third, experimental condition classification ‘C’ (listed in Table 4.7) involves varying the parameters that affect robots’ build quality and task risk levels. Finally, experimental condition classification ‘D’ (listed in Table 4.8) involves varying the initial resource amounts available in the simulation that are held by the robot system and the raw printing materials available in the environment.

Table 4.5. Experimental condition classification 'A' (robot cost).

ID	***Experimental Condition***	**Description**
A0	*(Default)*	Default values for all parameters.
A1	*BaseCost_Pr + 1*	BaseCost_Pr increased from 2 to 3.
A2	*BaseCost_Pr + 3*	BaseCost_Pr increased from 2 to 5.
A3	*BaseCost_Pr + 5*	BaseCost_Pr increased from 2 to 7.
A4	*PrintCost_Pr + 1*	PrintCost_Pr increased from 2 to 3.
A5	*PrintCost_Pr + 3*	PrintCost_Pr increased from 2 to 5.
A6	*PrintCost_Pr + 5*	PrintCost_Pr increased from 2 to 7.
A7	*AssembleCost_Pr + 1*	AssembleCost_Pr increased from 2 to 3.
A8	*AssembleCost_Pr + 3*	AssembleCost_Pr increased from 2 to 5.
A9	*AssembleCost_Pr + 5*	AssembleCost_Pr increased from 2 to 7.
A10	*BaseCost_Time + 2*	BaseCost_Time increased from 2 to 4.
A11	*BaseCost_Time - 1*	BaseCost_Time decreased from 2 to 1.
A12	*PrintCost_Time + 2*	PrintCost_Time increased from 2 to 4.
A13	*PrintCost_Time - 1*	PrintCost_Time decreased from 2 to 1.
A14	*AssembleCost_Time + 2*	AssembleCost_Time increased from 2 to 4.
A15	*AssembleCost_Time - 1*	AssembleCost_Time decreased from 2 to 1.
A16	*Print & Assemble Time + 2*	PrintCost_Time and AssembleCost_Time increased to 4.
A17	*BaseCost_NonPr + 1*	BaseCost_NonPr increased from 1 to 2.
A18	*PrintCost_NonPr + 1*	PrintCost_NonPr increased from 1 to 2.
A19	*AssembleCost_NonPr + 1*	AssembleCost_NonPr increased from 1 to 2.
A20	*[All]CostPrintable + 1*	Base-, Print-, and AssembleCost_Pr increased to 3.
A21	*[All]CostPrintable + 2*	Base-, Print-, and AssembleCost_Pr increased to 4.
A22	*Print & Assemble Pr + 2*	PrintCost_Pr and AssembleCost_Pr increased to 4.
A23	*Base & Print Pr + 2*	BaseCost_Pr and PrintCost_Pr increased to 4.
A24	*Base & Assemble Pr + 2*	BaseCost_Pr and AssembleCost_Pr increased to 4.
A25	*[All]CostPrintable - 1*	Base-, Print-, and AssembleCost_Pr decreased to 1.
A26	*AssembleCost_Pr - 1*	AssembleCost_Pr decreased from 2 to 1.
A27	*PrintCost_Pr - 1*	PrintCost_Pr decreased from 2 to 1.
A28	*BaseCost_Pr - 1*	BaseCost_Pr decreased from 2 to 1.

Table 4.6. Experimental condition classification 'B' (resource acquisition).

ID	***Experimental Condition***	**Description**
B1	*Print_Efficiency = 0.25*	Print_Efficiency decreased from 1.0 to 0.25.
B2	*Print_Efficiency = 0.5*	Print_Efficiency decreased from 1.0 to 0.5.
B3	*Print_Efficiency = 1.5*	Print_Efficiency increased from 1.0 to 1.5.
B4	*Collect_Amount = 0.25*	Collect_Amount decreased from 1.0 to 0.25.
B5	*Collect_Amount = 0.5*	Collect_Amount decreased from 1.0 to 0.5.
B6	*Collect_Amount = 1.5*	Collect_Amount increased from 1.0 to 1.5.
B7	*Print_Amount = 0.25*	Print_Amount decreased from 1.0 to 0.25.
B8	*Print_Amount = 0.5*	Print_Amount decreased from 1.0 to 0.5.
B9	*Print_Amount = 1.5*	Print_Amount increased from 1.0 to 1.5.
B10	*Collect & Print Amount = 0.5*	Collect_Amount and Print_Amount decreased to 0.5.

Table 4.7. Experimental condition classification 'C' (quality and risk).

ID	***Experimental Condition***	**Description**
C1	*QualityThreshold + 0.1*	QualityThreshold increased from 0.5 to 0.6.
C2	*QualityThreshold + 0.2*	QualityThreshold increased from 0.5 to 0.7.
C3	*QualityThreshold + 0.3*	QualityThreshold increased from 0.5 to 0.8.
C4	*QualityThreshold + 0.4*	QualityThreshold increased from 0.5 to 0.9.
C5	*RiskAmount_Print = 1%*	RiskAmount_Print increased from 0.1% to 1%.
C6	*RiskAmount_Assemble = 1%*	RiskAmount_Assemble increased from 0.1% to 1%.
C7	*RiskAmount Pr & As = 10%*	RiskAmount_Print & RiskAmount_Assemble = 10%.
C8	*RiskAmount Pr & As = 15%*	RiskAmount_Print & RiskAmount_Assemble = 15%.
C9	*RiskAmount_Assemble = 15%*	RiskAmount_Assemble increased from 0.1% to 15%.
C10	*Quality_incr_Chance = 0.01%*	Quality_incr_Chance decreased from 5% to 0.01%.
C11	*Quality_decr_Chance = 25%*	Quality_decr_Chance decreased from 50% to 25%.
C12	*Quality_decr_Chance = 75%*	Quality_decr_Chance increased from 50% to 75%.
C13	*Quality_decr_Upper = 0.5*	Quality_decr_Upper increased from 0.25 to 0.5.
C14	*Qual_incr Chance & Upper * 2*	Quality_incr_Chance = 10% & Quality_incr_Upper = 0.1.
C15	*RiskQuality_Modifier = 10.0*	RiskQuality_Modifier increased from 5.0 to 10.0.
C16	*RiskQuality_Modifier = 25.0*	RiskQuality_Modifier increased from 5.0 to 25.0.
C17	*RiskFactory_Modifier = 0.5*	RiskFactory_Modifier increased from 0.1 to 0.5.
C18	*RiskFactory_Modifier = 1.0*	RiskFactory_Modifier increased from 0.1 to 1.0.
C19	*Quality Thres & decr_Chance*	QualityThreshold = 0.9 & Quality_decr_Chance = 75%.

Table 4.8. Experimental condition classification 'D' (initial resources).

ID	***Experimental Condition***	**Description**
D1	*Initial_Printable / 2.0*	Initial_Printable decreased from 100 to 50.
D2	*Initial_Printable * 2.0*	Initial_Printable increased from 100 to 200.
D3	*Initial_Materials = 0*	Initial_Materials decreased from 50 to 0.
D4	*Initial_Materials / 2.0*	Initial_Materials decreased from 50 to 25.
D5	*Initial_Materials * 2.0*	Initial_Materials increased from 50 to 100.
D6	*Env_Materials / 2.0*	Env_Materials decreased from 500 to 250.
D7	*Env_Materials * 2.0*	Env_Materials increased from 500 to 1000.
D8	*Env_Materials * 100*	Env_Materials increased from 500 to 50000.
D9	*Initial_NonPr / 2.0*	Initial_NonPr decreased from 300 to 150.
D10	*Initial_NonPr * 2.0*	Initial_NonPr increased from 300 to 600.
D11	*Initial NonPr & Env * 2.0*	[D7, D10]
D12	*Initial NonPr & Env * 2.0, Raw=0*	[D3, D7, D10]

4.2.4. Hypotheses

There are three hypotheses that form the basis for the experiments presented in this chapter. There is a hypothesis specific to the centralized replication approach, one specific to the decentralized and hierarchical replication approaches, and one specific to the homogeneous and heterogeneous production approaches. Each is presented in this section.

4.2.4.1. Hypothesis 1: Centralized Replication Approach

The hypothesis for the centralized replication approach is that robot build quality will be higher than the decentralized and hierarchical replication approaches. The disadvantage of the centralized approach is that the assembly and printing capabilities are limited to the initial robots, and therefore its assembly and print potentials will be lower than the other replication approaches. Thus, the system collection potential and average build quality are the metrics of interest. The average build quality is expected to be higher for the centralized replication approach compared to the non-centralized replication approaches for all experimental conditions.

Systems using the centralized replication approach are not expected to have higher collection potential for the default case, since one assembling robot can only do so much in a given time period. In this regard, the experimental condition classification ‘C’ cases may have instances where the collection potential begins to favor the centralized replication approach in conditions with increased task risks and build quality demands. More specifically, the higher penalty experimental conditions C4, C7, C8, C9, C12, C13, and C19 are expected to show systems using the centralized approach being more competitive in terms of collection potential. Secondly, the costs associated with the assemble and print capability don’t affect the centralized approach (as only non-replicating robots are built). Thus, a significant enough increase in these parameters may make the centralized approach more competitive in terms of collection potential. The experimental conditions A6, A9, and A16 are examples where these parameters are significantly increased and are predicted to potentially have this occur.

Experimental condition C18 shows how the benefit of the centralized approach affects the outcome. This benefit is that only the initial robots are assembly capable. The initial robots are assumed to be well tested and therefore have a lower task risk level (simulated using

RiskFactory_Modifier). Thus, condition C18 shows the results of the system when that modifier is set to 1.0 (i.e., no task risk reduction for initial robots). It is expected that the collection potential would be lower due to the robots with an assembly capability succumbing to a task risk more often (on average), and therefore not being able to produce any more robots. The experimental condition C17 is expected to show a similar effect, but to a lesser extent (as it sets the modifier to 0.5).

4.2.4.2. Hypothesis 2: Decentralized versus Hierarchical

The hypothesis comparing the decentralized approach to the hierarchical approach is now discussed. The prediction is that the assembly and print potential will be higher in the decentralized approach as compared to the hierarchical approach. However, the collection potential for the decentralized approach is expected to be lower than that for the hierarchical approach, as systems using the decentralized approach do not build normal robots (i.e., lower production cost robots that can only collect). Furthermore, the production approach of the system (whether its homogeneous or heterogeneous), may impact performance in this area as well. Thus, it will primarily be of interest to compare the systems using the decentralized replication approach to their hierarchical approach counterpart (and vise-versa). In this regard, the DHE configuration would be compared to the HHE configuration and the DHO configuration would be compared to the HHO configuration.

The collection potential may be more equivalent between the decentralized and hierarchical replication approaches when the base production cost values of the robots are increased (and be farther apart when the base production costs are lowered). This prediction is due to the production cost of the normal robots being proportionally raised, which would therefore create a higher cost of building any robot. Experimental conditions A1, A2, and A3 affect the printable components' costs. A10 increases the required build duration, and A17 increases the nonprintable component costs for all robot types. Given this, it is predicted that under experimental conditions A1, A2, A3, A10, and A17, systems using the decentralized replication approach will have a more equivalent collection potential (as compared to the default case) to systems using the hierarchical replication approach. However, the nonprintable component cost increase under experimental condition A17 would only affect the system in the later stages (if it affects it at all) due to potentially causing the system to run out of this resource type.

Systems using the hierarchical replication approach may have a closer print and assembly potential to systems using the decentralized replication approach in instances where resource collection is more costly. Without enough resources, systems using the decentralized replication approach may not be able to produce robots as quickly. Because of this, it is estimated that under experimental conditions B4 and B5, systems using the hierarchical replication approach may be more competitive, in terms of assembly and print potential, as compared to systems using the decentralized approach. In addition, in the early stages (time-step 30), the experimental conditions D3 and D4 may show a similar effect. Experimental conditions B4 and B5 reduce the *Collect_Amount* parameter, which means that each robot collects less materials per time-step. Experimental conditions D3 and D4 involve reducing the amount of initial raw materials that the robot system begins the simulation with.

4.2.4.3. Hypothesis 3: Homogeneous versus Heterogeneous

The hypothesis for the comparison of the homogeneous and heterogeneous approaches is that the heterogeneous approach will be able to increase the number of robots more rapidly. This estimate is based on the heterogeneous production approach using specialized robots and having a lower production cost compared to the homogeneous approach (an exception to this is the CHE configuration, due to its reliance on initial robots). However, over time, the heterogeneous production approach would run out of nonprintable components. Therefore, it is predicted that systems using the homogeneous production approach will have a comparatively greater assembly potential and print potential at these later stages. This prediction is due to the homogeneous approach using the replicator robot type, which has both print and assemble capabilities. Therefore, fundamentally this test is a comparison of the value of versatility versus the value of build speed. Experimental conditions D6 and D9 are predicted to have this occur earlier, in terms of time-steps, due to there being less available resources in these experimental conditions.

Another mission planning consideration is the relative production costs of the various robot types (both the resources and build time required). Having an asymmetrical cost for the print capability and assemble capability is considered, as the base cost affects both roughly equally. Because of this, it is predicted that experimental conditions A4, A5, A6, A7, A8 and A9 will result in the DHE and HHE configurations outperforming the DHO and HHO configurations in terms of collect, print, and assembly potential.

The parameters *RiskAmount_Print* and *RiskAmount_Assemble* may also influence the results. These may affect the homogeneous and heterogeneous production approaches differently. Experimental conditions C5 and C6 are predicted to result in the DHO and HHO configurations outperforming the DHE and HHE configurations in terms of print and assembly potential. This is because the replicator robots have both assemble and print capabilities and may be less affected by losing one of the two capabilities. However, the potential for an assemble task to fail, while producing a more resource expensive replicator robot, may be a counter balancing drawback. The higher risk level under experimental conditions C7, C8, and C9 may show a more pronounced effect, and equally impair the DHO, HHO, DHE, and HHE configurations.

4.3. Results

In this section, the results of the experiments performed are summarized. Full result tables for the primary metrics at time-steps 30, 50, and 70 are provided in Appendix A. In Table 4.9, the results for the default case of A0 (which sets all parameters at their default values) are listed for each system configuration and time-step.

Table 4.9. Results for each system configuration on the default case (A0).

A0	**Assembly Potential**			**Print Potential**			**Collection potential**		
Values	**30**	**50**	**70**	**30**	**50**	**70**	**30**	**50**	**70**
CHE	0.99	1.00	1.00	1.00	0.98	1.00	13.77	20.25	29.21
CHO	1.00	1.00	1.00	1.00	1.00	1.00	12.84	18.81	28.24
DHE	22.67	69.88	73.82	18.36	58.22	62.10	41.33	129.48	137.41
DHO	15.70	38.93	82.95	15.66	38.81	82.99	15.78	39.28	83.85
HHE	18.35	63.27	63.79	13.57	53.07	53.81	44.19	164.48	165.53
HHO	13.56	32.19	70.91	13.67	32.19	70.67	23.19	66.46	142.48

An overview of the results for each system configuration, across an entire classification of experimental conditions, is provided in Tables 4.11-4.14. These tables report the percent share of the total of each column in terms of the sum of each system configuration across the experimental condition classification. Due to the variance of each experimental condition in the experimental condition classification, this percentage may be skewed toward high-performing experimental conditions over low-performing experimental conditions.

Table 4.10 shows the percentage breakdown of the values in Table 4.9 (default case). Table 4.11 shows this for experimental condition classification '*A*' (28 experimental conditions). Table 4.12 shows this for experimental condition classification '*B*' (10 experimental conditions). Table 4.13 shows this for experimental condition classification '*C*' (19 experimental conditions). Table 4.14 shows this for experimental condition classification '*D*' (12 experimental conditions).

Table 4.10. Percentage of total for each system configuration for the default case (A0).

A0	Assembly Potential			Print Potential			Collection Potential		
Percentage	**30**	**50**	**70**	**30**	**50**	**70**	**30**	**50**	**70**
CHE	1.37%	0.48%	0.34%	1.58%	0.53%	0.37%	9.11%	4.62%	4.98%
CHO	1.38%	0.48%	0.34%	1.58%	0.54%	0.37%	8.50%	4.29%	4.81%
DHE	31.37%	33.88%	25.15%	29.02%	31.59%	22.87%	27.35%	29.51%	23.42%
DHO	21.72%	18.87%	28.27%	24.75%	21.06%	30.56%	10.44%	8.95%	14.29%
HHE	25.39%	30.67%	21.74%	21.45%	28.80%	19.81%	29.25%	37.49%	28.21%
HHO	18.76%	15.61%	24.16%	21.61%	17.47%	26.02%	15.35%	15.15%	24.28%

Table 4.11. Percentage of total for experimental condition classification ‘*A*’.

Sum of Classification ‘A’	Assembly Potential			Print Potential			Collection Potential		
	30	**50**	**70**	**30**	**50**	**70**	**30**	**50**	**70**
CHE	1.56%	0.64%	0.45%	1.88%	0.76%	0.53%	10.29%	6.10%	6.26%
CHO	1.56%	0.64%	0.45%	1.89%	0.76%	0.53%	9.54%	5.73%	5.71%
DHE	31.16%	32.78%	28.97%	26.72%	29.47%	25.41%	25.26%	27.03%	24.05%
DHO	21.84%	21.27%	25.68%	26.37%	25.13%	30.25%	10.33%	9.93%	12.20%
HHE	26.38%	28.86%	25.27%	21.97%	25.19%	20.69%	28.44%	32.94%	28.64%
HHO	17.49%	15.80%	19.18%	21.18%	18.68%	22.58%	16.13%	18.28%	23.15%

Table 4.12. Percentage of total for experimental condition classification '*B*'.

Sum of Classification 'B'	**Assembly Potential**			**Print Potential**			**Collection Potential**		
	30	**50**	**70**	**30**	**50**	**70**	**30**	**50**	**70**
CHE	1.44%	0.66%	0.45%	1.65%	0.77%	0.52%	9.53%	6.29%	6.12%
CHO	1.43%	0.66%	0.45%	1.65%	0.77%	0.52%	8.85%	6.01%	5.91%
DHE	29.98%	32.00%	28.12%	27.66%	29.29%	24.80%	26.23%	27.51%	24.13%
DHO	22.36%	21.90%	25.71%	25.70%	25.48%	29.73%	10.82%	10.50%	12.49%
HHE	25.21%	27.17%	23.52%	20.77%	23.18%	19.23%	28.49%	31.87%	28.20%
HHO	19.59%	17.62%	21.75%	22.58%	20.52%	25.20%	16.08%	17.82%	23.15%

Table 4.13. Percentage of total for experimental condition classification '*C*'.

Sum of Classification 'C'	**Assembly Potential**			**Print Potential**			**Collection potential**		
	30	**50**	**70**	**30**	**50**	**70**	**30**	**50**	**70**
CHE	1.71%	0.61%	0.41%	1.97%	0.70%	0.46%	10.47%	5.50%	5.70%
CHO	1.70%	0.62%	0.41%	2.03%	0.74%	0.49%	9.60%	5.22%	5.41%
DHE	31.31%	33.02%	26.19%	28.24%	30.37%	23.54%	26.75%	28.46%	23.78%
DHO	21.12%	19.47%	26.69%	24.73%	22.01%	29.15%	10.23%	9.29%	13.23%
HHE	25.28%	30.33%	22.53%	20.86%	28.05%	20.32%	27.94%	36.42%	28.43%
HHO	18.89%	15.95%	23.76%	22.17%	18.12%	26.04%	15.02%	15.11%	23.45%

Table 4.14. Percentage of total for experimental condition classification '*D*'.

Sum of Classification 'D'	**Assembly Potential**			**Print Potential**			**Collection potential**		
	30	**50**	**70**	**30**	**50**	**70**	**30**	**50**	**70**
CHE	1.41%	0.52%	0.35%	1.63%	0.58%	0.38%	9.17%	4.92%	4.79%
CHO	1.41%	0.52%	0.35%	1.63%	0.58%	0.38%	8.56%	4.60%	4.57%
DHE	31.36%	32.76%	27.47%	28.75%	30.31%	25.71%	26.96%	28.23%	25.18%
DHO	21.34%	20.21%	25.98%	24.73%	22.46%	27.75%	10.23%	9.48%	12.66%
HHE	25.93%	29.85%	23.46%	21.69%	28.13%	21.96%	29.24%	36.46%	29.98%
HHO	18.55%	16.15%	22.39%	21.57%	17.95%	23.83%	15.83%	16.32%	22.81%

4.3.1. Robot Build Rate Comparison

In this subsection, the robot build rate for each system configuration under experimental condition A0 (the default case) is presented. Figure 4.1 depicts the build rate for the DHE configuration. Figure 4.2 depicts the build rate for the DHO configuration. Figure 4.3 depicts the build rate for the HHE configuration, and Figure 4.4 depicts the build rate for the HHO configuration. Figure 4.5 depicts the build rate for the CHE and CHO configurations.

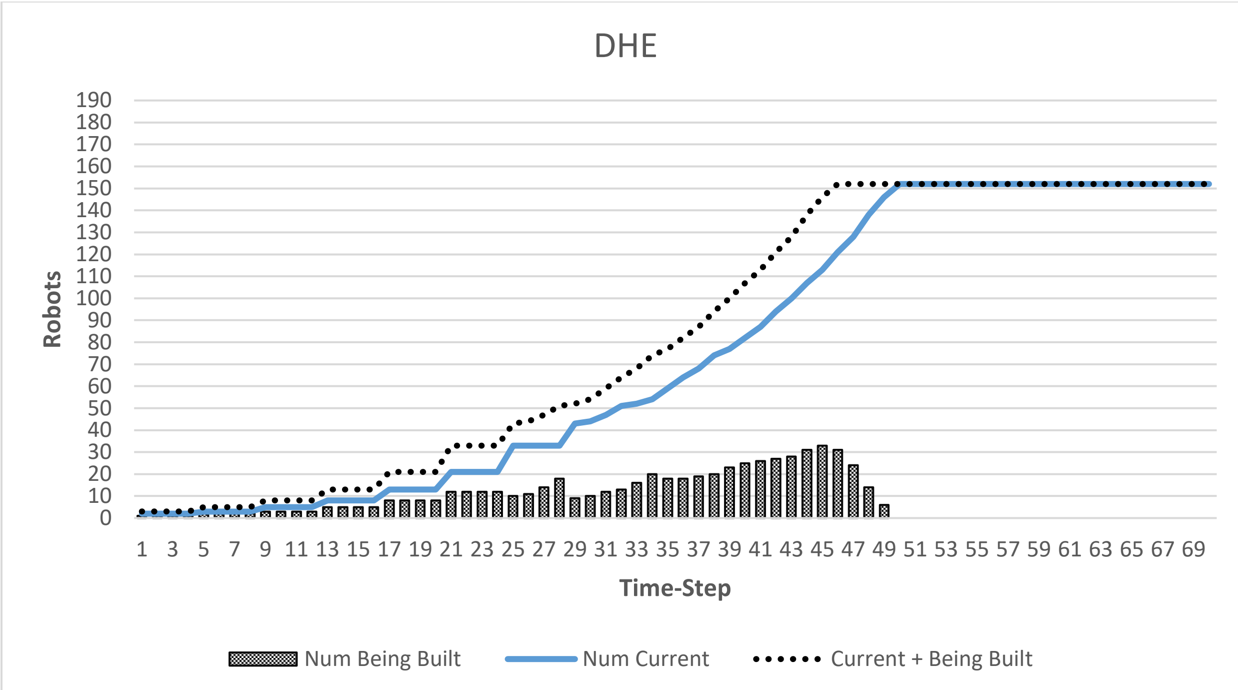


Figure 4.1. Robot build rate of the DHE configuration on experimental condition A0.

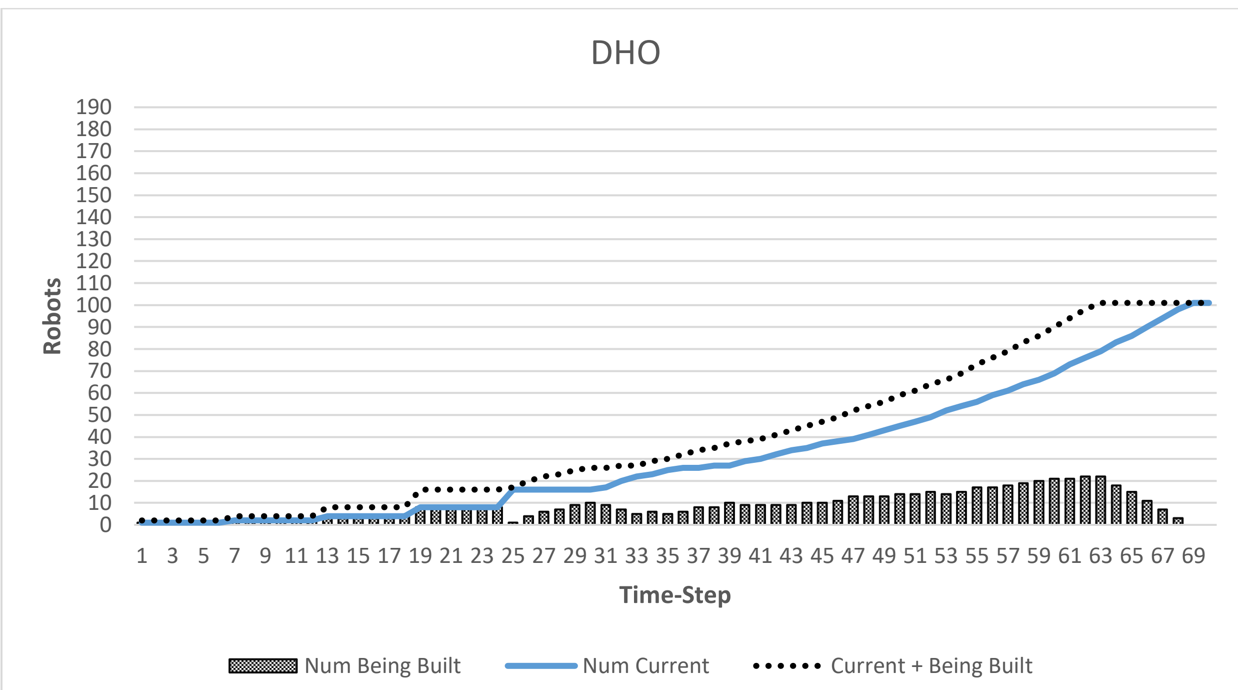


Figure 4.2. Robot build rate of the DHO configuration on experimental condition A0.

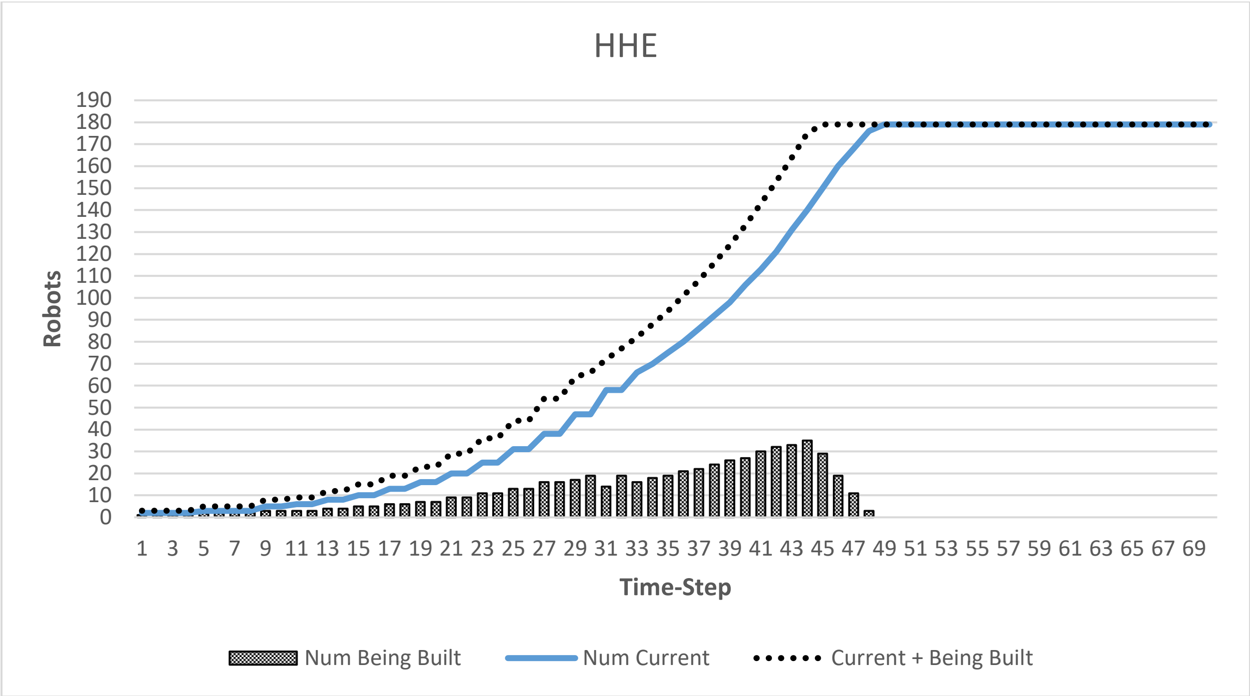


Figure 4.3. Robot build rate of the HHE configuration on experimental condition A0.

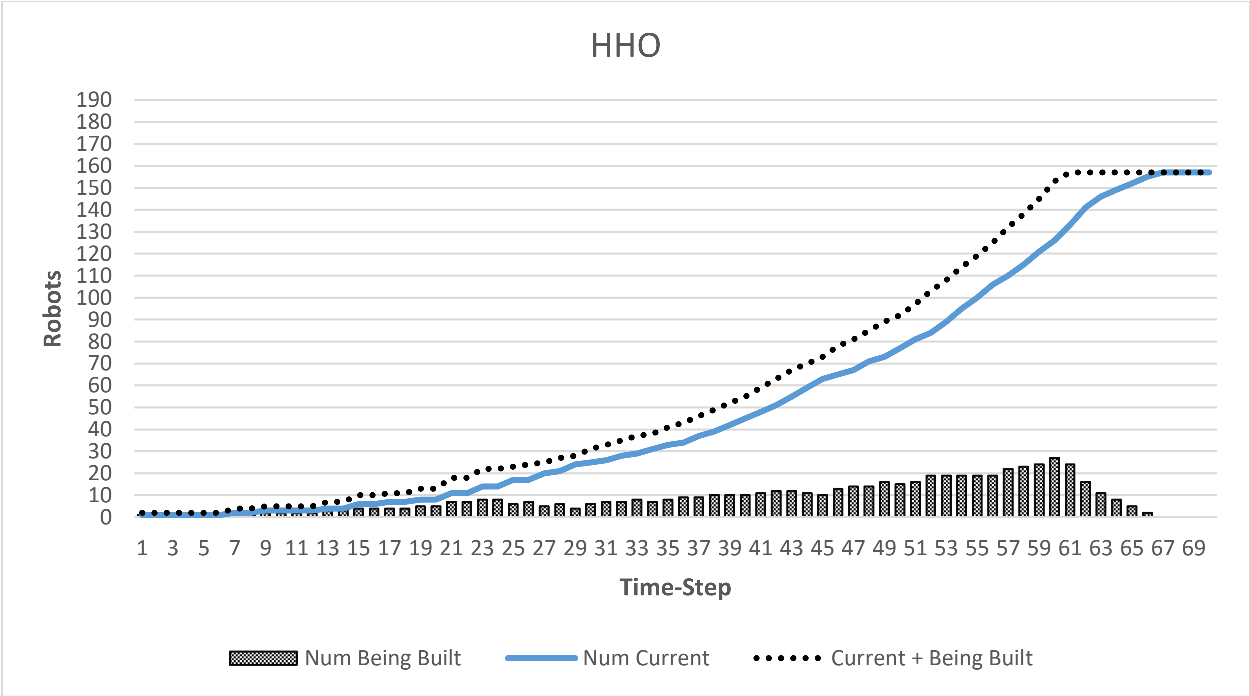


Figure 4.4. Robot build rate of the HHO configuration on experimental condition A0.

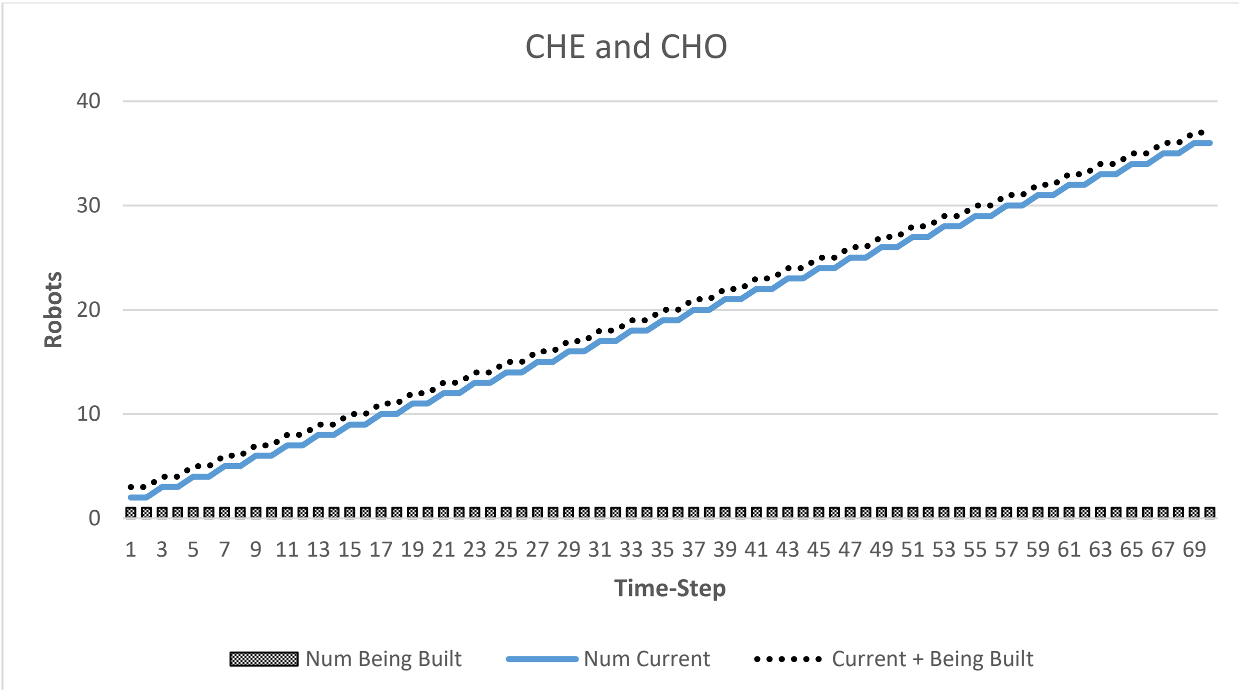


Figure 4.5. Robot build rate of the CHE and CHO configurations on experimental condition A0.

4.4. Analysis

In this section, the results are analyzed to determine if the hypotheses are supported or refuted by the data.

4.4.1. Evaluation of Hypothesis 1: Centralized Replication Approach

The average build quality was predicted to be higher for the system configurations using the centralized replication approach (CHE and CHO) as compared to the other replication approaches. Analysis supports the hypothesis of average robot build quality being higher for systems using the centralized approach. An average of all experimental conditions for each system configuration for the metric of average robot build quality is presented in Table 4.15, and the individual results for each experimental condition are provided in Table A.13 (Appendix A).

Table 4.15. Average robot build quality across all experimental conditions.

Average	CHE	CHO	DHE	DHO	HHE	HHO
Time-Step: 30	0.945	0.942	0.875	0.887	0.876	0.875
Time-Step: 50	0.942	0.940	0.854	0.873	0.852	0.855
Time-Step: 70	0.941	0.939	0.850	0.860	0.850	0.846

In terms of collection potential, experimental conditions A6 and A9 did not lower the production of robots for most of the non-centralized configurations enough to make the collection potential of these systems more comparable to the centralized configurations CHE and CHO. An exception is the DHO configuration, which was lowered enough to be roughly equivalent to the CHE and CHO configurations in terms of collection potential at time-step 30 (Table 4.16) and 50 (Table 4.17). Furthermore, the collection potential of the CHE and CHO configurations was close to standard deviation range of the DHO configuration at time-step 70 (Table 4.18).

Table 4.16. Centralized hypothesis-related data for time-step 30.

(30)	***Experimental Condition***	**Collection Potential**					
		CHE	**CHO**	**DHE**	**DHO**	**HHE**	**HHO**
A0	*(Default)*	13.77	12.84	41.33	15.78	44.19	23.19
A6	*PrintCost_Pr + 5*	13.42	12.67	31.96	11.77	39.31	26.25
A9	*AssembleCost_Pr + 5*	13.93	12.83	26.57	11.75	33.99	25.83
A16	*Print & Assemble Time + 2*	13.73	12.57	12.73	3.95	17.26	6.61
C4	*QualityThreshold + 0.4*	10.72	9.30	21.23	8.20	21.75	11.12
C7	*RiskAmount Pr & As = 10%*	12.24	11.07	14.15	6.79	13.93	8.70
C8	*RiskAmount Pr & As = 15%*	11.32	10.45	10.92	5.00	10.71	7.03
C9	*RiskAmount_Assemble = 15%*	11.87	10.35	10.67	5.04	11.53	6.95
C12	*Quality_decr_Chance = 75%*	13.40	12.34	39.71	15.26	42.87	22.26
C13	*Quality_decr_Upper = 0.5*	13.15	12.57	35.69	12.96	38.06	20.12
C17	*RiskFactory_Modifier = 0.5*	13.81	12.72	41.68	15.73	43.42	23.16
C18	*RiskFactory_Modifier = 1.0*	13.41	12.26	40.87	15.54	43.41	22.88
C19	*Quality Thres & Chance*	8.73	7.23	12.64	5.21	12.04	7.22

Table 4.17. Centralized hypothesis-related data for time-step 50.

(50)	Collection Potential						*Collection Potential Std Dev*					
	CHE	**CHO**	**DHE**	**DHO**	**HHE**	**HHO**	**CHE**	**CHO**	**DHE**	**DHO**	**HHE**	**HHO**
A0	20.25	18.81	129.48	39.28	164.48	66.46	*2.07*	*1.98*	*11.67*	*3.15*	*7.92*	*3.83*
A6	19.88	19.24	71.88	19.72	97.06	63.05	*2.11*	*2.58*	*5.00*	*2.14*	*5.12*	*4.35*
A9	19.96	18.77	57.51	19.20	80.82	62.44	*2.29*	*2.07*	*5.44*	*2.70*	*7.47*	*4.60*
A16	19.43	18.79	54.32	15.43	70.05	27.61	*2.27*	*2.06*	*3.19*	*0.82*	*5.00*	*2.54*
C4	15.02	14.79	53.19	20.79	64.19	30.73	*2.58*	*2.32*	*12.13*	*4.93*	*16.30*	*9.35*
C7	16.01	14.89	22.62	11.48	23.07	15.44	*6.71*	*6.76*	*9.99*	*5.86*	*10.40*	*8.41*
C8	14.68	12.54	13.89	7.75	15.13	10.50	*6.71*	*6.93*	*7.19*	*4.14*	*8.47*	*5.99*
C9	13.39	13.55	16.45	9.10	15.17	10.65	*6.63*	*6.58*	*8.95*	*4.96*	*9.79*	*5.74*
C12	19.23	18.83	116.78	37.37	154.73	60.54	*2.16*	*1.75*	*16.05*	*3.89*	*10.31*	*8.66*
C13	19.10	18.02	96.00	32.29	126.82	53.09	*2.28*	*2.17*	*16.01*	*5.33*	*21.26*	*7.68*
C17	19.53	18.70	128.11	39.37	161.86	65.10	*3.29*	*2.38*	*12.15*	*2.92*	*18.17*	*5.21*
C18	18.83	18.38	126.89	38.79	163.25	64.46	*4.18*	*2.65*	*12.45*	*4.73*	*17.89*	*7.91*
C19	12.14	11.40	27.43	12.44	30.75	18.61	*2.04*	*2.48*	*7.83*	*4.09*	*11.14*	*6.48*

Table 4.18. Centralized hypothesis-related data for time-step 70.

(70)	Collection Potential						*Collection Potential Std Dev*					
	CHE	**CHO**	**DHE**	**DHO**	**HHE**	**HHO**	**CHE**	**CHO**	**DHE**	**DHO**	**HHE**	**HHO**
A0	29.21	28.24	137.41	83.85	165.53	142.48	*2.85*	*2.19*	*6.86*	*8.50*	*6.04*	*6.29*
A6	29.54	28.08	110.92	31.99	159.15	105.12	*2.84*	*3.33*	*7.52*	*4.07*	*24.72*	*6.57*
A9	28.68	28.17	89.92	31.94	115.32	105.25	*2.86*	*3.60*	*3.64*	*3.38*	*7.68*	*7.86*
A16	29.23	28.02	136.69	34.41	162.49	65.89	*3.28*	*2.62*	*7.47*	*2.65*	*6.65*	*7.01*
C4	20.44	19.13	90.84	34.33	111.84	64.04	*4.01*	*3.09*	*13.12*	*9.22*	*9.01*	*18.44*
C7	20.18	19.63	27.74	12.39	27.14	21.88	*11.03*	*11.13*	*12.98*	*7.09*	*15.75*	*13.75*
C8	18.29	14.95	14.47	7.75	16.26	12.28	*10.82*	*11.32*	*9.28*	*4.59*	*9.60*	*8.39*
C9	17.86	16.50	17.22	9.25	16.88	12.25	*11.09*	*10.88*	*10.59*	*6.68*	*12.55*	*9.09*
C12	28.69	27.19	129.48	75.31	156.02	133.79	*2.65*	*3.48*	*8.48*	*11.56*	*8.99*	*9.89*
C13	28.42	27.10	117.49	61.40	142.8	115.38	*2.72*	*2.89*	*10.95*	*13.29*	*12.22*	*17.27*
C17	28.69	27.48	138.77	84.03	165.39	142.07	*3.09*	*3.75*	*5.62*	*8.74*	*5.58*	*6.56*
C18	28.01	26.54	138.39	84.24	165.67	143.67	*5.05*	*7.48*	*4.78*	*8.59*	*6.16*	*5.66*
C19	15.69	14.48	50.78	19.77	58.54	33.24	*3.43*	*3.51*	*16.58*	*6.97*	*21.04*	*12.46*

For experimental conditions C4 and C19, the homogeneous configurations DHO and HHO performed similar to the centralized configurations in terms of collection potential at the early stages (time-step 30). In the later stages, the HHO configuration was notably higher than the centralized configurations under experimental conditions C4 and C19, while the DHO configuration's performance remained comparable. In contrast, experimental condition C12 did not result in a significantly different collection potential, as compared to the default case, and C13 did not lower the collection potential of the other configurations enough to be similar in value to the CHE and CHO configurations.

The higher risk experimental conditions (C7, C8, and C9) lowered the collection potential of the non-centralized configurations enough such that the CHE and CHO configurations were competitive. This appears to be primarily attributable to the non-factory built assemble-capable robots having too high a task risk level to assemble, to a functional level, properly in these cases. In comparison, the experimental condition C4 corresponds to having a high rate of build failure (the robot construction failed). Under experimental conditions C7, C8, and C9, the dynamic is slightly different in that the risk involves the assembling robot failing during the assembly process and thus losing the assemble capability. In this case, not only does the build fail, but the assembling robot also loses its capability. The results support the hypothesis that the non-centralized configurations are, to some extent, able to better cope with a high rate of build failure but are less able to cope with a high rate of assembly equipment failure.

Finally, experimental conditions C17 and C18 did not show a significant impairment of collection potential, as compared to the default case. With a 0.1% chance of the assembly robot losing the assemble capability per time-step (due to the task risk level), failure did not occur enough to have a significant impact.

4.4.2. Evaluation of Hypothesis 2: Decentralized Versus Hierarchical

The descriptions of the experimental conditions used to evaluate this hypothesis are listed in Table 4.19. The results of condition A0 (default case) support the hypothesis that the hierarchical configurations have a higher collection potential compared to their decentralized counterparts.

Furthermore, the decentralized configurations had an increased print and assembly potential compared to their hierarchical counterpart. This supports the hypothesis as well.

Table 4.19. Description of experimental conditions for hypothesis 2.

ID	Experimental Condition	Description
A0	*(Default)*	Default values for all parameters.
A1	*BaseCost_Pr + 1*	BaseCost_Pr increased from 2 to 3.
A2	*BaseCost_Pr + 3*	BaseCost_Pr increased from 2 to 5.
A3	*BaseCost_Pr + 5*	BaseCost_Pr increased from 2 to 7.
A10	*BaseCost_Time + 2*	BaseCost_Time increased from 2 to 4.
A17	*BaseCost_NonPr + 1*	BaseCost_NonPr increased from 1 to 2.
B4	*Collect_Amount = 0.25*	Collect_Amount decreased from 1.0 to 0.25.
B5	*Collect_Amount = 0.5*	Collect_Amount decreased from 1.0 to 0.5.
D3	*Initial_Materials = 0*	Initial_Materials decreased from 50 to 0.
D4	*Initial_Materials / 2.0*	Initial_Materials decreased from 50 to 25.

Experimental conditions A1, A2, and A3 affect the *Base_CostPr* parameter on an increasing scale. For these three experimental conditions, the decentralized configurations were closer, in terms of collection potential, to the hierarchical configurations than the default case (shown in Table 4.20). Unexpectedly, the DHE configuration outperformed the HHE configuration under experimental condition A2 at the time-steps of 50 and 70. In addition, it outperformed for experimental condition A3 by a marginal amount at time-step 50 (and was approximately equivalent to the highest performing conditions at time-steps 30 and 70). This supports the hypothesis that raising the base production cost of robots (in terms of printable components) increases the comparative performance of decentralized configurations, compared to hierarchical configurations, in terms of collection potential. Although, it is more prevalent for the DHE configuration, as compared to the HHE configuration, and more marginal of a comparative gain for the DHO configuration versus the HHO configuration.

Experimental condition A10 involves increasing the *BaseCost_Time* parameter, which affects how long it takes to build all robot types. While this was hypothesized to potentially favor the collection potential of decentralized configurations over hierarchical configurations, the results showed a pattern similar to a slowly progressing default case (A0). Experimental condition A17 increases the nonprintable component cost for all robots (*BaseCost_NonPr*), and the results indicate that this didn't significantly enhance the comparative performance of the decentralized approach, as compared to the hierarchical approach.

Table 4.20. Collection potential and standard deviation for certain experimental conditions.

ID	**Collection Potential**				**Collection Potential Std Dev**			
	DHE	**DHO**	**HHE**	**HHO**	**DHE**	**DHO**	**HHE**	**HHO**
	Time-Step: 30							
A0	41.33	15.78	44.19	23.19	*2.33*	*0.52*	*4.96*	*1.59*
A1	32.89	15.63	39.33	22.23	*1.59*	*0.71*	*2.01*	*1.88*
A2	26.34	13.52	26.79	18.51	*1.18*	*1.03*	*1.62*	*1.31*
A3	19.27	11.74	19.87	13.31	*1.16*	*0.54*	*1.33*	*0.84*
A10	12.74	7.92	15.55	10.44	*0.69*	*0.37*	*0.89*	*0.90*
A17	41.58	15.53	44.65	23.04	*2.04*	*1.11*	*1.88*	*1.90*
	Time-Step: 50							
A0	129.4	39.28	164.4	66.46	*11.67*	*3.15*	*7.92*	*3.83*
A1	94.52	33.40	114.8	53.66	*12.34*	*2.66*	*8.97*	*4.17*
A2	53.32	24.71	40.87	32.92	*5.97*	*2.27*	*2.84*	*2.34*
A3	28.39	19.93	26.87	27.19	*2.57*	*1.99*	*2.52*	*1.84*
A10	54.78	24.96	53.74	30.26	*3.67*	*1.42*	*3.45*	*2.27*
A17	90.59	39.15	100.3	65.49	*4.69*	*3.06*	*4.99*	*4.00*
	Time-Step: 70							
A0	137.4	83.85	165.5	142.4	*6.86*	*8.50*	*6.04*	*6.29*
A1	120.9	64.89	136.8	118.3	*4.89*	*7.09*	*4.86*	*6.75*
A2	81.21	43.74	60.85	70.66	*9.35*	*4.90*	*3.65*	*5.65*
A3	38.96	31.35	37.59	46.97	*4.35*	*3.92*	*3.71*	*5.95*
A10	137.4	49.71	161.3	74.59	*8.61*	*4.36*	*9.53*	*5.91*
A17	91.32	65.36	102.3	90.65	*4.54*	*3.33*	*4.28*	*4.57*

Experimental conditions B4 and B5 involve reducing the *Collect_Amount* parameter, where B4 reduces it by a factor of four and B5 reduces it by half. These two experimental conditions affected the performance of the decentralized configurations, as compared to the hierarchical configurations in terms of assembly potential and print potential (shown in Table 4.21). The HHE configuration was approximately equal to the DHE configuration in terms of print and assembly potentials under experimental conditions B4 and B5 at time-step 50. However, under experimental condition B5, this converged to approximately the same as the default case by time-step 70, presumably due to reaching the maximum number of robots within resource constraints. The HHO configuration approximately equaled the DHO configuration at time-step 50 and notably outperformed it at time-step 70. These results differ from the default case, where the DHO configuration outperformed the HHO configuration, and the DHE configuration outperformed the HHE configuration.

In contrast, experimental conditions D3 and D4 did not significantly impact the relative performance of decentralized configurations compared to hierarchical configurations in terms of assembly potential and print potential. These experimental conditions did, though, marginally impact the DHO configuration (but not the DHE configuration) at time-steps 50 and 70.

Table 4.21. Assembly and print potentials for relevant experimental conditions.

ID	Assembly Potential				Print Potential			
	DHE	DHO	HHE	HHO	DHE	DHO	HHE	HHO
	Time-Step: 30							
A0	22.67	15.70	18.35	13.56	18.36	15.66	13.57	13.67
B4	21.37	15.64	17.88	13.54	16.22	15.68	12.80	13.68
B5	21.71	15.44	18.51	13.66	17.36	15.45	13.09	13.72
D3	19.90	14.72	18.41	13.69	14.05	14.85	13.12	13.76
D4	20.82	15.20	18.38	13.55	16.53	15.27	13.52	13.63
	Time-Step: 50							
A0	69.88	38.93	63.27	32.19	58.22	38.81	53.07	32.19
B4	35.77	27.58	33.34	27.21	25.67	27.66	26.15	27.19
B5	50.36	32.61	49.53	31.55	37.97	32.72	38.59	31.68
D3	61.08	32.42	59.32	31.91	48.43	32.46	48.97	31.91
D4	65.55	36.50	63.05	32.02	54.61	36.58	52.18	31.96
	Time-Step: 70							
A0	73.82	82.95	63.79	70.91	62.10	82.99	53.81	70.67
B4	59.86	36.52	56.97	42.05	42.11	36.65	44.21	42.20
B5	72.88	55.79	62.25	68.92	57.16	55.65	49.88	68.98
D3	76.38	70.16	64.35	70.62	61.34	70.10	52.42	70.50
D4	74.91	77.87	63.54	70.91	62.96	78.02	53.38	70.74

4.4.3. Evaluation of Hypothesis 3: Heterogeneous Versus Homogeneous

The descriptions of the experimental conditions used in the evaluation of this hypothesis are listed in Table 4.22. The results of the default case (A0) support the hypothesis that the (heterogeneous) DHE and HHE configurations produce robots more rapidly than the (homogeneous) DHO and HHO configurations. Under this experimental condition, the DHE and HHE configurations had reached the resource-constraint-based maximum number of robots by time-step 50 (depicted in Figures 4.1 and 4.3). In contrast, the DHO and HHO configurations had a more linear build rate, which reached the maximum number of robots around time-step 70 (depicted in Figures 4.2 and 4.4). In this regard, the DHO and HHO configurations outperform their heterogeneous counterpart in terms of assembly and print potentials by time-step 70. This supports the hypothesis that the homogeneous production approach would have a greater assembly potential and print potential than the heterogeneous production approach at later time-steps.

Table 4.22. Description of experimental conditions for hypothesis 3.

ID	Experimental Condition	Description
A0	*(Default)*	Default values for all parameters.
A4	*PrintCost_Pr + 1*	PrintCost_Pr increased from 2 to 3.
A5	*PrintCost_Pr + 3*	PrintCost_Pr increased from 2 to 5.
A6	*PrintCost_Pr + 5*	PrintCost_Pr increased from 2 to 7.
A7	*AssembleCost_Pr + 1*	AssembleCost_Pr increased from 2 to 3.
A8	*AssembleCost_Pr + 3*	AssembleCost_Pr increased from 2 to 5.
A9	*AssembleCost_Pr + 5*	AssembleCost_Pr increased from 2 to 7.
C5	*RiskAmount_Print = 1%*	RiskAmount_Print increased from 0.1% to 1%.
C6	*RiskAmount_Assemble = 1%*	RiskAmount_Assemble increased from 0.1% to 1%.
D6	*Env_Materials / 2.0*	Env_Materials decreased from 500 to 250.
D9	*Initial_NonPr / 2.0*	Initial_NonPr decreased from 300 to 150.

In terms of assembly potential, the DHE and HHE configurations continued to outperform their homogeneous counterparts at time-step 70 under experimental conditions A5, A6, A8, and A9 (as shown in Table 4.23). Similarly, in terms of print potential, the DHE and HHE configurations outperformed under experimental conditions A5, A8, and A9 (but not for A6) at time-step 70.

The experimental conditions A4 and A7, which increase printable cost to a lesser extent, resulted in the DHE and HHE configurations performing closer (as compared to A0) to their homogeneous counterpart at time-step 70, for both assembly and print potentials. Thus, the increase in the printable component cost appears to comparatively favor the heterogeneous production approach over the homogeneous production approach. This provides support for the hypothesis that the heterogeneous production approach is able to continue to outperform the homogeneous approach, in terms of assembly and print potentials, for these experimental conditions, even when robot numbers reach a resource constrained maximum.

Table 4.23. Results of the DHE, DHO, HHE, and HHO configurations for conditions A4 to A9.

ID	Assembly Potential				Print Potential				Collection Potential			
	DHE	**DHO**	**HHE**	**HHO**	**DHE**	**DHO**	**HHE**	**HHO**	**DHE**	**DHO**	**HHE**	**HHO**
	Time-Step: 30											
A0	22.67	15.70	18.35	13.56	18.36	15.66	13.57	13.67	41.3	15.8	44.2	23.2
A4	20.23	15.38	18.33	12.77	16.43	15.43	13.59	12.79	37.0	15.5	43.6	23.2
A5	19.67	13.57	18.02	10.86	12.96	13.55	12.64	10.88	32.9	13.6	40.7	22.6
A6	21.24	11.75	16.87	8.89	10.46	11.73	10.61	8.94	32.0	11.8	39.3	26.3
A7	19.73	15.40	18.36	12.84	16.45	15.37	13.60	12.86	36.5	15.5	44.0	23.4
A8	16.71	13.59	16.96	10.76	14.14	13.55	10.86	10.82	31.2	13.7	38.4	22.1
A9	14.97	11.73	12.94	8.79	11.33	11.63	11.80	8.91	26.6	11.8	34.0	25.8
	Time-Step: 50											
A0	69.88	38.93	63.27	32.19	58.22	38.81	53.07	32.19	129.5	39.3	164.5	66.5
A4	63.96	32.79	58.35	28.63	49.52	32.74	48.30	28.70	114.7	33.1	157.0	62.0
A5	52.84	25.08	47.18	11.81	36.22	24.88	35.05	11.92	90.1	25.2	122.6	72.4
A6	53.66	19.64	51.73	8.76	17.49	19.49	10.56	8.81	71.9	19.7	97.1	63.1
A7	57.25	33.15	57.76	28.25	49.53	32.98	51.31	28.16	108.1	33.3	155.7	61.1
A8	41.04	24.50	38.25	11.60	40.86	24.50	33.49	11.69	83.0	24.8	103.3	71.9
A9	30.11	19.04	29.67	8.71	26.67	19.04	27.31	8.76	57.5	19.2	80.8	62.4
	Time-Step: 70											
A0	73.82	82.95	63.79	70.91	62.10	82.99	53.81	70.67	137.4	83.9	165.5	142.5
A4	74.39	64.88	63.08	64.86	58.28	64.81	52.29	64.59	134.1	65.6	166.3	142.6
A5	68.39	43.22	57.32	12.28	46.18	43.09	42.55	12.21	115.7	43.6	145.4	129.9
A6	81.63	31.83	87.07	8.60	27.88	31.49	10.42	8.62	110.9	32.0	159.2	105.1
A7	70.58	65.45	61.10	64.63	61.38	65.50	53.65	64.50	133.4	66.2	161.6	143.0
A8	53.52	44.33	50.21	12.08	53.50	44.26	45.33	12.15	108.3	44.6	135.0	130.9
A9	44.90	31.69	42.78	8.68	44.07	31.48	38.49	8.69	89.9	31.9	115.3	105.3

Under experimental condition D9, the DHO and HHO configurations were more equal to DHE and HHE (as compared to A0), in terms of assembly and print potentials, at time-step 50 (as shown in Table 4.24). Moreover, the DHO configuration already outperformed the DHE and HHE configurations for these metrics at time-step 50. In the default case, this happened at a later time-step. Experimental condition D9 supported the hypothesis that a configuration using the homogeneous production approach would overtake one using the heterogeneous production approach at an earlier time-step. Under experimental condition D6, the performance of the homogeneous configurations was closer to the heterogeneous configurations, as compared to the default case. The DHO configuration outperformed, in terms of print potential, for this experimental condition at time-step 50, but it was still marginally lower, in terms of assembly potential. While not outperforming in terms of both print and assembly potentials for experimental condition D6, it still provides limited support for the hypothesis that homogeneous configurations would outperform heterogeneous configurations at earlier time-steps.

Table 4.24. Results of DHE, DHO, HHE, and HHO on select experimental conditions.

ID	Assembly Potential				Print Potential			
	DHE	**DHO**	**HHE**	**HHO**	**DHE**	**DHO**	**HHE**	**HHO**
	Time-Step: 30							
A0	22.67	15.70	18.35	13.56	18.36	15.66	13.57	13.67
C5	22.65	15.51	18.30	13.54	17.39	14.91	12.30	13.26
C6	20.17	12.25	14.35	11.42	17.19	13.47	11.28	12.46
D6	22.72	15.37	18.44	13.70	18.78	15.46	13.56	13.76
D9	22.56	15.57	18.59	13.76	18.42	15.60	13.71	13.81
	Time-Step: 50							
A0	69.88	38.93	63.27	32.19	58.22	38.81	53.07	32.19
C5	68.48	38.90	58.65	30.95	50.03	36.46	41.12	27.90
C6	54.06	32.56	52.65	26.35	49.10	35.31	49.28	29.57
D6	52.84	39.20	44.59	31.97	42.11	39.17	35.76	31.86
D9	34.72	39.17	32.11	31.96	30.67	39.04	26.56	31.95
	Time-Step: 70							
A0	73.82	82.95	63.79	70.91	62.10	82.99	53.81	70.67
C5	74.97	81.29	64.50	70.31	55.53	73.98	45.09	62.55
C6	65.19	64.16	54.20	58.08	59.30	70.22	51.31	65.23
D6	52.30	59.44	44.80	48.24	42.24	59.77	36.04	48.43
D9	35.24	40.38	32.25	36.19	31.02	40.10	26.46	35.96

Experimental condition C6, which increases the assemble task risk level, resulted in the opposite of what was predicted. The results of experimental condition C6 indicate that the heterogeneous approach outperformed the homogeneous approach. Compared to the default case, the DHE, DHO, HHE, and HHO configurations were more equal, in terms of assembly potential, and were reduced approximately equally, in terms of print potential, at time-step 70. Earlier time-steps did not significantly show either the homogeneous or heterogeneous production approaches outperforming the other.

Experimental condition C5, which increases the print task risk level, did not result in performance that deviated significantly from the results of the default case in terms of assembly potential. There was a slight deviation, in terms of print potential; however, this did not result in superior performance from the homogeneous configurations. The high-risk experimental conditions (C7, C8, and C9) did not show significant differences in the comparative performance of the homogeneous or heterogeneous approaches. Thus, the hypothesis of the parameters *RiskAmount_Print* and *RiskAmount_Assemble* affecting the comparative performance of the hierarchical approach versus the homogeneous approach was not supported by the results.

4.5. Summary

In this chapter, an experiment was conducted using the simulation system presented in Chapter 3 to study the differences in performance caused by the system configurations of CHE, CHO, DHE, DHO, HHE, and HHO. The centralized configurations CHE and CHO were shown to only marginally outperform in high-risk cases in terms of collection potential, as compared to other configurations. The heterogeneous configurations of DHE and HHE (not configuration CHE) were shown to reach a maximum number of robots (based on available resources) more quickly than their homogeneous configuration counterparts. However, the homogeneous configurations (DHO and HHO) outperformed them in terms of assembly potential and print potential in later time-steps (for many experimental conditions). Finally, the decentralized configurations (DHE and DHO) outperformed their hierarchical counterpart in terms of assembly potential and print potential for most experimental conditions; however, they had a lower collection potential. Whether a system was a homogeneous or heterogeneous configuration was shown to be a more significant factor overall. These findings, along with the data acquired from the experiment in this chapter, were used to establish the decision-making criteria for the decision-making algorithms. These decision-making algorithms are presented in the next chapter.

5. DECISION-MAKING EXPERIMENT[5]

In this chapter, the decision-making algorithms devised from the results from the work presented in Chapter 4 are discussed. The decision-making algorithm used for a system will depend on the system's mission objectives. For instance, maximizing the number of robots with a particular capability may be a requirement for a mission. This might be the case if the end goal was to print a large structure and a number of 3D print-capable robots are needed. Martian habitat creation using in situ material 3D printing [18] is an example of one use for this type of capability. Similarly, a mission involving assembling a large structure, which will be assembled from pre-fabricated components, could necessitate maximizing the number of assembly-capable robots. In contrast, a more even distribution of robot types (or of robots with differing capabilities) may be needed for other missions with multiple objectives or uncertain conditions.

The performance of these algorithms is compared to the results of the base algorithm. The presented algorithms are also compared to each other. First, a review of the operation of the base algorithm is provided. Then, the cycle decision-making algorithm is discussed. Third, the variable decision-making algorithm is presented. Last, the strategic decision-making algorithm is detailed. The results of each of these decision-making algorithms are provided and analyzed.

5.1. Base Decision-Making Algorithm

In the base decision-making algorithm, the choice of when to build a new robot and what type it should be is decided with simple criteria. The algorithm finds all idle assembly-capable robots and takes the buildable robot type list (see Table 5.1) and choose the next type from the list,

---

[5] This chapter is derived from: A. Jones and J. Straub, " Evaluation of Algorithms for Heuristic Decision-Making for 3D Printed Self-Replicating Robots," *(submitted to) J. Intell. Robot. Syst.*, 2020.

repeating the list, in order, until there are enough robots in the build queue for each idle assembly-capable robot. Under this algorithm, resource constraints are not checked initially. If a robot cannot be built, the would-be assembler simply remains idle.

Table 5.1. Buildable robot types for each system configuration.

Buildable Robot Types	Centralized	Decentralized	Hierarchical
Homogeneous	Normal	Replicator	Replicator, Normal
Heterogeneous	Normal	Assembler, Printer	Assembler, Printer, Normal

After determining what each of the robots need to assemble (if anything), the base decision-making algorithm assigns all currently idle print-capable robots to fabricate printable components. This is limited by the robot system's current amount of available raw printing materials (robots will not be assigned to printing tasks that materials are not available for).

After these assignments, all robots that are idle are assigned to collect materials from the environment. If all of the robots return no materials during a given time-step, then it is assumed that the environment is out of raw materials and the system stops assigning robots to the collection task once this happens. The task risk associated with collection motivates discontinuing unfruitful collection. However, the stage at which it can be assumed that no further resources are available to collect may be more complex in real world instances.

5.2. Cycle Decision-Making Algorithm

In the cycle decision-making algorithm, the process for choosing when to build a new robot and what type it should be is very similar to the approach used in the base algorithm. The algorithm begins by finding all idle assembly-capable robots and sequentially assigning them robot types to build from the buildable robot type list (see Table 5.1). Ordered assignment is repeated until each idle assembly-capable robot has been assigned a task.

The difference between the cycle algorithm and the base algorithm is that the position in the build order that is used for assignment is retained from the previous time-step (shown in Figure 5.1). Thus, the base algorithm will always restart at the first robot type in the list, which means it will favor the first robot type early on (when there are a limited number of assemble-capable robots). It will also slightly favor robot types earlier in the list throughout all operations. Retaining the position in the build order ensures that the ratio of robot types in the system is more balanced. However, this only makes a difference for systems' behavior when the system has more than one buildable robot type. The CHE, CHO, and DHO configurations are therefore unaffected by the differences between the two. The versions of the DHE, HHE, and HHO configurations utilizing the cycle algorithm are denoted with the prefix: "Cycle-" (i.e., Cycle-HHE).

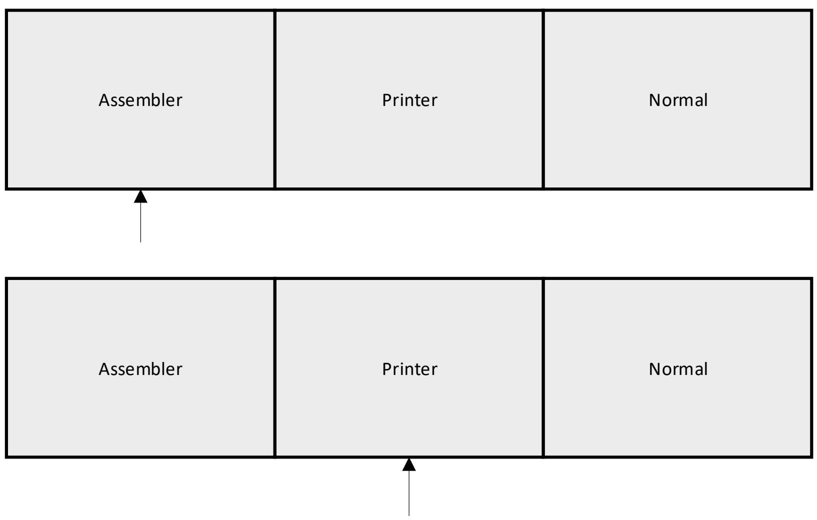


Figure 5.1. Diagram of the cycle decision-making algorithm.

The process of assigning robots to the print and collect tasks under the cycle algorithm is identical to the base algorithm. The cycle decision-making algorithm assigns all currently idle print-capable robots to fabricate new printable components. Assignments are limited by the robot system's current amount of available raw printing materials as robots will not be assigned to printing tasks that materials are not available for. After printing assignments, all robots that are still idle are assigned to collect materials from the environment. The system stops sending robots out to collect resources based on the same criteria as the base algorithm.

5.2.1. Hypothesis

The performance of the cycle algorithm is predicted to differ from the base algorithm in the early time-steps. This prediction is because, once there are enough assemblers and resources, the build list may be long enough such that the difference between the algorithms has little impact on cycling in the later time-steps. Specifically, it is expected that the cycle decision-making algorithm will have a higher print potential for the DHE and HHE configurations in early stages (as compared to the base algorithm), and a minor increased potential in later stages. The HHO configuration is excluded from this prediction because of the use of replicator robots. This makes the print and assembly potentials equal. Instead, the cycle-HHO configuration is predicted to have a slightly increased collection potential as compared to the base algorithm, as more normal robots may be constructed.

Table 5.2. Experimental conditions where DHE, HHE, and HHO had atypical ratio results.

Time-Step (70)	**ID**	**Assemble Ratio**			**Print Ratio**			**Assembly Potential**			**Print Potential**		
Experimental Condition		DHE	HHE	HHO	DHE	HHE	HHO	DHE	HHE	HHO	DHE	HHE	HHO
(Default)	**A0**	0.54	0.39	0.49	0.45	0.32	0.49	74.6	64.6	70.1	62.9	53.4	70.4
BaseCost_Pr + 5	**A3**	0.85	0.59	0.70	0.14	0.13	0.70	34.1	21.4	31.3	5.7	4.6	31.0
PrintCost_Pr + 5	**A6**	0.74	0.53	0.08	0.25	0.06	0.08	81.8	84.1	8.5	27.3	9.7	8.8
AssembleCost_Pr + 5	**A9**	0.50	0.37	0.08	0.48	0.35	0.08	44.7	42.4	8.3	42.9	40.1	8.7
[All]CostPrintable + 2	**A21**	0.85	0.58	0.12	0.14	0.11	0.12	40.9	34.9	7.9	6.5	6.4	7.7
Base & Print Pr + 2	**A23**	0.88	0.66	0.12	0.11	0.09	0.12	67.9	51.6	8.7	8.1	7.4	8.4
Print_Efficiency = 0.25	**B1**	0.76	0.35	0.20	0.22	0.15	0.19	39.5	24.2	14.5	11.7	10.3	14.1
Print_Amount = 0.25	**B7**	0.76	0.35	0.20	0.23	0.15	0.20	41.6	24.1	14.5	12.2	10.1	14.4

Based on results from the base algorithm (chapter 4), under certain experimental conditions the assemble ratio and print ratio significantly differed from the typical values (as observed in most other conditions) for the DHE, HHE, and HHO configurations. The experimental conditions where this was most pronounced were: A3, A6, A9, A21, A23, B1, and B7. The values of the assemble ratio, print ratio, assembly potential, and print potential for these cases are listed in Table 5.2.

The DHE and HHE configurations were less affected by experimental conditions A6 and A9 and may remain relatively unchanged when using the cycle algorithm. The more significant changes are predicted to occur under experimental conditions A3, A21, A23, B1, and B7. Under these experimental conditions, it is predicted that the assemble and print ratios will be more balanced for the DHE and HHE configurations when using the cycle algorithm. In addition, it is predicted that the assembly potential will be lower, and the print potential will be higher for the DHE and HHE configurations, as compared to their performance with the base algorithm.

For the HHO configuration, the use of the cycle algorithm may be less favorable, in terms of both the assembly potential and the print potential. The problem that the base algorithm HHO configuration may have been facing, under experimental conditions A6, A9, A21 and A23, is that the printable component cost of replicator robots was significantly higher than the cost of normal robots. Under these conditions, more normal robots will be built regardless of the build order, as the robot system wouldn't have enough resources to build the replicator robots. In experimental conditions B1 and B7, a similar issue may be caused by printable component acquisition. This results in the same dynamic as the previously discussed experimental conditions, since the limited components will be assigned to producing the lower production cost normal robots. Thus, the hypothesis for the Cycle-HHO configuration is that it will underperform, as compared to the base algorithm, in terms of assembly potential and print potential in these experimental conditions; however, it may outperform in terms of collection potential.

5.3. Variable Decision-Making Algorithm

The variable decision-making algorithm (VAR) is less constrained than the base and cycle algorithms. It is able to change system configurations during a simulation run. The variable algorithm uses a system configuration of the same name – "variable". This system configuration is capable of building any robot type. The algorithm alters its buildable robot list to match other system configurations, based on which one is deemed to be the most favorable to the input parameters. Different system configurations may be more favorable for different criteria, so an input is needed as to which attribute is desirable to optimize for. The attributes/metrics that are selected to be optimized for by the algorithm include collection potential, assembly potential, and print potential.

The ability to switch to a known high performing configuration (in terms of an input metric) functions as a heuristic. The switches represent a greedy algorithm that chooses the configuration based on data-driven values to improve the expected performance.

The choice of system configuration is based on the results from Chapter 4 (base algorithm), as well as the performance of the cycle algorithm. Specifically, the input metric to optimize for (i.e., collection potential) is used with the analysis of the simulation parameters known to the robot system to determine the system configuration that is estimated to perform the best in the applicable conditions. A 'lookup' table is used for this purpose. Since the input metric will affect its performance, the variable algorithm cases are denoted as follows:

- **Variable-A (*VAR-A*)**: variable decision-making algorithm with an inputted optimization metric of assembly potential.
- **Variable-C (*VAR-C*)**: variable decision-making algorithm with an inputted optimization metric of collection potential.

- **Variable-P (*VAR-P*)**: variable decision-making algorithm with an inputted optimization metric of print potential.

The variable decision-making algorithm can switch system configurations at any given time-step. This is dependent on which time-steps there are result data for, so the time-steps at which the switches happen in the experimental runs are time-steps 30, 50, and 70. The first system configuration is selected based on the best performance (for the input metric) at time-step 30. Upon reaching time-step 30, the system configuration is switched to the one that performs best at time-step 50. The same is done for the transition at time-step 50. If the same system configuration is estimated to perform best at a given interval, then no switch occurs (i.e., it would switch to the same configuration, resulting in no change).

Only the simulation parameters that are known to the robot system are used to switch system configurations. Of course, it is not possible to plan based on unknown simulation parameters. A list of which simulation parameters are known and which are unknown to the robot system is provided in Table 5.3. For the experimental conditions where unknown simulation parameters are altered from default values, the starting system configuration selected is the one that optimizes for the base case (or whichever case best fits the known parameters). Then, the current state of the robot system is compared with the data from the lookup table and switched if the state aligns with data from an experimental condition where an unknown parameter was altered. Thus, the first possible switch due to unknown parameters could happen at time-step 30, for cases where the unknown parameters have been changed.

Table 5.3. Parameters known (visible) to the robot system.

Known	Unknown
Num_Steps	*QualityThreshold*
Initial_NonPr	*Quality_incr_Chance*
Initial_Printable	*Quality_incr_Lower*
Initial_Materials	*Quality_incr_Upper*
Env_Materials	*Quality_decr_Chance*
BaseCost_NonPr	*Quality_decr_Lower*
BaseCost_Pr	*Quality_decr_Upper*
BaseCost_Time	*RiskAmount_Collect*
PrintCost_NonPr	*RiskAmount_Print*
PrintCost_Pr	*RiskAmount_Assemble*
PrintCost_Time	*RiskQuality_Modifier*
AssembleCost_NonPr	*RiskFactory_Modifier*
AssembleCost_Pr	
AssembleCost_Time	
Print_Efficiency	
Print_Amount	
Collect_Amount	

The process for assigning robots to the print and collect tasks is identical to the base and cycle algorithms. All currently idle print-capable robots are assigned to fabricate new printable components. The ability to do this is limited by the robot system's current amount of available raw printing materials (robots are not assigned to printing tasks that materials are not available for). After these assignments, all robots that are idle are assigned to collect materials from the environment. The system stops sending robots out to collect resources based on the same criteria as the base and cycle algorithms.

5.3.1. Lookup Charts

The build list lookup charts contain the information as to which system configuration the variable decision-making algorithm should change to, and at what stage. The lookup chart for experimental conditions with classification ‘A’ is presented in Table 5.4. The list for experimental conditions with classification ‘B’, ‘C’, and ‘D’ are listed in Tables 5.5, 5.6, and 5.7, respectively. The stages listed in the lookup charts are as follows:

- **Early**: Starts at timestep 0 and ends at timestep 30.
- **Mid**: Starts at timestep 30 and ends at timestep 50.
- **Late**: Starts at timestep 50 and ends at timestep 70.

Experimental condition classification ‘C’ involves parameters unknown to the robot system, so initially the system is set to optimize for the default case. The lookup chart assumes that a distinguishable pattern will be evident by the mid stage.

Table 5.4. Build chart for experimental condition classification 'A'.

ID	Assembly			Collection			Print		
	Early	Mid	Late	Early	Mid	Late	Early	Mid	Late
A0	DHE	DHE	DHO	HHE	HHE	Cycle-HHE	DHE	Cycle-DHE	DHO
A1	DHE	DHE	DHO	HHE	Cycle-HHE	Cycle-HHE	Cycle-DHE	Cycle-DHE	DHO
A2	DHE	DHE	DHE	Cycle-HHE	Cycle-HHE	Cycle-HHE	DHO	Cycle-DHE	DHO
A3	DHE	DHE	DHE	Cycle-HHE	Cycle-DHE	Cycle-DHE	DHO	Cycle-DHE	Cycle-DHE
A4	DHE	DHE	DHE	HHE	HHE	HHE	Cycle-DHE	Cycle-DHE	HHO
A5	DHE	DHE	DHE	HHE	HHE	Cycle-HHE	Cycle-DHE	Cycle-DHE	Cycle-DHE
A6	DHE	HHE	HHE	HHE	Cycle-HHE	HHE	DHO	Cycle-DHE	Cycle-DHE
A7	DHE	HHE	DHE	HHE	HHE	Cycle-HHE	Cycle-DHE	HHE	Cycle-DHE
A8	HHE	DHE	DHE	HHE	Cycle-HHE	Cycle-HHE	Cycle-DHE	Cycle-DHE	Cycle-DHE
A9	DHE	DHE	DHE	HHE	Cycle-HHE	Cycle-HHE	Cycle-DHE	Cycle-DHE	Cycle-DHE
A10	DHE	DHE	DHE	HHE	DHE	HHE	DHO	DHO	Cycle-DHE
A11	DHE	DHE	DHO	Cycle-HHE	Cycle-HHE	Cycle-HHE	Cycle-DHE	Cycle-DHE	DHO
A12	DHE	DHE	DHE	HHE	HHE	HHE	DHE	Cycle-DHE	Cycle-DHE
A13	DHE	DHE	DHO	HHE	HHE	Cycle-HHE	DHO	Cycle-DHE	DHO
A14	DHE	DHE	DHE	HHE	HHE	Cycle-HHE	Cycle-DHE	Cycle-DHE	Cycle-DHE
A15	DHE	DHE	DHO	HHE	HHE	HHE	Cycle-DHE	Cycle-DHE	DHO
A16	DHE	DHE	DHE	HHE	HHE	HHE	Cycle-HHE	Cycle-DHE	DHE
A17	DHE	DHE	DHO	HHE	HHE	Cycle-HHE	Cycle-DHE	Cycle-DHE	DHO
A18	DHE	DHE	DHO	HHE	HHE	HHE	DHE	Cycle-DHE	DHO
A19	DHE	DHE	DHO	HHE	Cycle-HHE	HHE	DHE	Cycle-DHE	DHO
A20	DHE	DHE	DHE	HHE	Cycle-HHE	Cycle-HHE	Cycle-DHE	Cycle-DHE	Cycle-DHE
A21	DHE	DHE	DHE	Cycle-HHE	Cycle-HHE	Cycle-HHE	Cycle-DHE	Cycle-DHE	Cycle-DHE
A22	DHE	DHE	DHE	HHE	Cycle-HHE	Cycle-HHE	Cycle-DHE	Cycle-DHE	Cycle-DHE
A23	DHE	DHE	DHE	HHE	Cycle-HHE	Cycle-HHE	DHO	Cycle-DHE	Cycle-DHE
A24	DHE	DHE	DHE	Cycle-HHE	Cycle-HHE	Cycle-HHE	Cycle-DHE	Cycle-DHE	Cycle-DHE
A25	DHE	DHO	DHO	DHE	HHE	Cycle-HHE	DHE	DHO	DHO
A26	DHE	DHE	DHO	DHE	HHE	Cycle-HHE	DHE	Cycle-DHE	DHO
A27	DHE	DHE	DHO	DHE	HHE	Cycle-HHE	DHE	DHE	DHO
A28	DHE	DHE	DHO	DHE	HHE	Cycle-HHO	DHE	Cycle-DHE	DHO

Table 5.5. Build chart for experimental condition classification 'B'.

ID	**Assembly**			**Collection**			**Print**		
	Early	**Mid**	**Late**	**Early**	**Mid**	**Late**	**Early**	**Mid**	**Late**
B1	DHE	DHE	DHE	HHE	Cycle-HHE	HHO	DHO	Cycle-DHE	DHO
B2	DHE	DHE	HHE	HHE	Cycle-HHE	Cycle-HHE	Cycle-DHE	Cycle-DHE	HHO
B3	DHE	DHE	DHO	DHE	HHE	Cycle-HHE	DHE	Cycle-DHE	DHO
B4	DHE	DHE	DHE	HHE	Cycle-HHE	Cycle-HHE	Cycle-DHE	Cycle-DHE	Cycle-HHE
B5	DHE	HHE	DHE	HHE	HHE	Cycle-HHE	Cycle-DHE	Cycle-DHE	HHO
B6	DHE	DHE	DHO	HHE	HHE	Cycle-HHE	Cycle-DHE	Cycle-DHE	DHO
B7	DHE	DHE	DHE	HHE	Cycle-HHE	Cycle-HHE	Cycle-DHE	DHO	Cycle-DHE
B8	DHE	DHE	DHE	HHE	Cycle-DHE	Cycle-HHE	Cycle-DHE	Cycle-DHE	Cycle-DHE
B9	DHE	DHE	DHO	HHE	HHE	HHE	DHE	DHE	DHO
B10	DHE	DHE	DHE	HHE	Cycle-HHE	Cycle-HHE	Cycle-DHE	Cycle-DHE	Cycle-DHE

Table 5.6. Build chart for experimental condition classification 'C'.

ID	**Assembly**			**Collection**			**Print**		
	Early	**Mid**	**Late**	**Early**	**Mid**	**Late**	**Early**	**Mid**	**Late**
C1	DHE	DHE	DHO	HHE	HHE	Cycle-HHE	DHE	DHE	DHO
C2	DHE	DHE	DHE	HHE	HHE	HHE	DHE	DHE	DHO
C3	DHE	DHE	DHE	HHE	HHE	HHE	DHE	DHE	DHO
C4	DHE	DHE	DHE	HHE	HHE	HHE	DHE	Cycle-DHE	Cycle-DHE
C5	DHE	DHE	DHO	HHE	HHE	Cycle-HHE	DHE	Cycle-DHE	DHO
C6	DHE	DHE	DHO	HHE	HHE	HHE	DHE	DHE	DHO
C7	DHE	DHE	DHO	HHE	DHE	HHO	DHE	DHO	HHO
C8	DHE	DHO	DHE	HHE	CHE	CHE	DHE	DHO	DHO
C9	DHE	HHO	DHE	HHE	CHE	DHE	DHE	HHO	DHO
C10	DHE	DHE	DHO	HHE	HHE	HHE	DHE	DHE	DHO
C11	DHE	DHE	DHO	HHE	HHE	Cycle-HHE	DHE	Cycle-DHE	DHO
C12	DHE	DHE	DHO	HHE	HHE	Cycle-HHE	DHE	Cycle-DHE	DHO
C13	DHE	DHE	DHE	HHE	HHE	HHE	DHE	HHE	HHO
C14	DHE	DHE	DHO	HHE	HHE	Cycle-HHE	DHE	DHE	DHO
C15	DHE	DHE	DHO	HHE	HHE	Cycle-HHE	DHE	Cycle-DHE	DHO
C16	DHE	HHE	HHO	HHE	HHE	Cycle-HHE	DHE	HHE	HHO
C17	DHE	DHE	DHO	HHE	HHE	HHE	DHE	DHE	DHO
C18	DHE	DHE	DHO	HHE	HHE	HHE	DHE	DHE	DHO
C19	DHE	DHE	DHE	HHE	HHE	HHE	DHE	DHO	Cycle-DHE

Table 5.7. Build chart for experimental condition classification 'D'.

ID	**Assembly**			**Collection**			**Print**		
	Early	**Mid**	**Late**	**Early**	**Mid**	**Late**	**Early**	**Mid**	**Late**
D1	DHE	DHE	DHE	HHE	Cycle-HHE	Cycle-HHE	Cycle-DHE	Cycle-DHE	Cycle-DHE
D2	DHE	DHE	DHO	DHE	HHE	Cycle-HHE	DHE	Cycle-DHE	DHO
D3	DHE	DHE	DHE	HHE	HHE	Cycle-HHE	Cycle-DHE	Cycle-DHE	DHO
D4	DHE	DHE	DHO	HHE	HHE	HHE	Cycle-DHE	Cycle-DHE	DHO
D5	DHE	DHE	DHO	DHE	HHE	HHE	DHE	Cycle-DHE	DHO
D6	DHE	DHE	DHO	HHE	Cycle-HHE	HHE	DHE	Cycle-DHE	DHO
D7	DHE	DHE	DHO	HHE	HHE	HHE	DHE	Cycle-DHE	DHO
D8	DHE	DHE	DHO	HHE	HHE	Cycle-HHE	DHE	DHE	DHO
D9	DHE	DHO	DHO	HHE	Cycle-HHE	Cycle-HHE	DHE	DHO	DHO
D10	DHE	DHE	DHO	HHE	HHE	HHE	DHE	DHE	DHO
D11	DHE	DHE	DHE	HHE	HHE	HHE	Cycle-DHE	Cycle-DHE	Cycle-DHE
D12	DHE	DHE	DHE	HHE	Cycle-HHE	HHE	Cycle-DHE	Cycle-DHE	Cycle-DHE

5.3.2. Hypothesis

Since the algorithm has its first possible decision-making point at time-step 30, comparing its results at that stage would be the same as comparing the first item in its lookup chart at time-step 30. Thus, instead of measuring at time-step 30 for the variable decision-making algorithm, the measurement is taken at time-step 40 instead. Therefore, the time-steps that results are collected for are: 40, 50, and 70.

The variable decision-making algorithm is predicted to have the highest performance, in terms of its selected metric for optimization, at time-steps 40 and 50 for most experimental conditions (as compared to the base and cycle decision-making algorithms). In certain experimental conditions where heterogeneous configurations are favored early on, the robot system may run out of resources. Thus, the heterogeneous configurations are expected to reach capacity faster and be outperformed in the later stages by the homogeneous configurations. Based on the data presented in chapter 4, this would only affect the print potential and assembly potential. Thus, the variable-P and variable-A approaches are predicted to have potentially lower performance, with regards to their respective metric, at time-step 70 under experimental conditions A11, A13, A15, A17, A18, A19, A25, A26, A27, A28, B3, B6, B9, D2, D5, D6, and D9. Under these experimental conditions, the maximum number of robots (based on resource constraints) was reached in earlier time-steps (with the base algorithm) as compared to the base case of A0.

In terms of collection potential, the variable-C approach is predicted to have the highest (or be tied for highest) in terms of collection potential for all of the experimental conditions and time-steps. Ties are predicted because, in certain experimental conditions, the variable-C approach has the same system configuration for all of its stages. In these cases, in effect it is the same as that system configuration (or close to it). The possibility of ties due to this also applies to the variable-P and variable-A approaches in regard to their respective optimization metric.

Second, the variable decision-making algorithm is predicted to slightly outperform the base and cycle decision-making algorithms for the high-risk experimental conditions C7, C8, and C9. This is predicted because the system is predicted to switch configuration in order to adapt to the higher task risk levels. However, the increase is not anticipated to be significant, and in some cases no increase may occur.

5.4. Strategic Decision-Making Algorithm

The strategic decision-making algorithm (STR) is even less constrained than the variable decision-making algorithm. Similar to the variable algorithm, it also uses a variable system configuration. It adjusts the configuration according to what is estimated to perform the best for the given circumstances. However, the strategic algorithm will use any build list of robots that it projects that it needs. Therefore, it isn’t constrained to known system configurations. The strategic decision-making algorithm also utilizes an input attribute/metric that is desired to be optimized for. For the experimental runs, the input optimization metrics are the same as used in the variable algorithm: collection potential (STR-C), print potential (STR-P), and assembly potential (STR-A). In addition, the algorithm also utilizes a time-step input goal. For the experimental runs, this is the *Num_Steps* parameter.

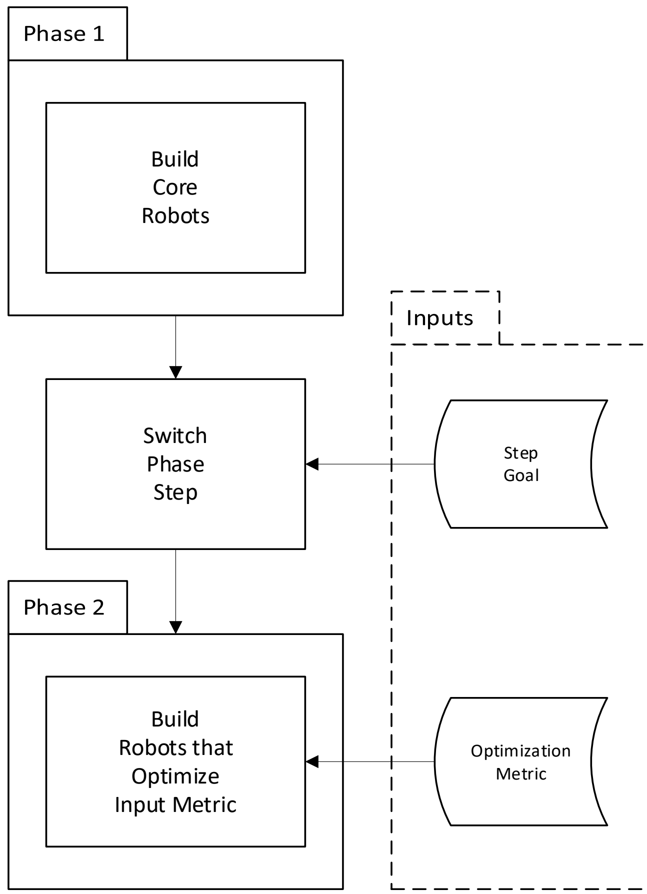


Figure 5.2. Diagram of the overall operation of the strategic decision-making algorithm.

For the strategic decision-making algorithm, the overall plan of which robot types to build is divided into two phases. The first phase involves building up a 'core' number of robots which have (or are projected to have) the needed capabilities to build the robots for the second phase. The second phase consists of building robots targeted to increase performance in terms of the input optimization metric. For example, if the optimization metric is print potential, then the second phase would consist of only building printer robots and/or replicator robots, since they have the requisite print capability. An overview of this process is depicted in Figure 5.2.

The choice of which robot types to build for the first phase is dependent on the optimization metric, the time-step goal, and the known simulation parameters. For instance, if the input metric is print potential, then the robot types of the core robots would be selected to ensure that adequate assemble-capable and collect-capable robots are built. Thus, in phase 1 the metric that is optimized for is based on providing a utility in later stages (i.e., in phase two). In phase one (early stage), the focus is on producing a sufficient quantity of robots with the desired capabilities.

The time-step goal (i.e., at which step the simulation stops) affects which robots are built in phase one as well. For instance, if the goal is in the later stages (i.e., step 70), then (based on the data presented in chapter 4) it may be more beneficial to build replicator robots as compared to assembler or printer robots.

The time-step goal affects the point at which the phase switches from one to two. The time-steps where the switch takes place between phase one and phase two are listed in Table 5.8. These were decided as follows. Based on the data presented in chapter 4, time-step 30 is a point where the systems configurations had used approximately a third or less of their available nonprintable components. In comparison, at time-step 50 certain system configurations had utilized all of their available nonprintable components. Switching to phase two without any nonprintable components remaining would be counterproductive. Thus, for the purposes of the experiment, the time-step at which to switch to phase two will be set to time-step 30. However, in the case where the step goal is 30, switching then is too late – so the time-step at which to switch for this case was set to be time-step 15.

Table 5.8. Step at which to switch to phase 2.

Step Goal	**30**	**50**	**70**
Switch to Phase 2	*15*	*30*	*30*

The algorithm used for assigning robots to the print and collection tasks is the same as for the base and cycle algorithms. The system assigns all currently idle print-capable robots to fabricate new printable components. Assignment is limited by the robot system's current amount of available raw printing materials, as robots will not be assigned to printing tasks that materials are not available for. After these assignments are made, all robots that are idle are assigned to collect materials from the environment. Furthermore, this algorithm stops sending robots out to collect resources based on the same criteria as the base and cycle algorithms.

5.4.1. Initial Build Chart

The initial build chart for the strategic algorithm is listed in Table 5.9. cycle-DHE algorithm was chosen to use to build the core robots for the strategic-A algorithm due to its high early-stage print potential. This is needed when the system is mostly building assembly-capable robots. This was also the rationale for choosing it for phase 1 of the strategic-C algorithm. However, when attempting to maximize for the print potential metric, as is the case with strategic-P, it is necessary to build assembly-capable robots in phase one. Because of this, DHE was chosen to provide the increased assembly potential to the strategic-P algorithm in phase 1.

Table 5.9. Initial build chart.

Listing	**Input Metric**	**Phase 1**			**Phase 2**
		Step Goal=30	Step Goal =50	Step Goal =70	
Strategic-C	*Collection potential*	Cycle-DHE	Cycle-DHE	Cycle-DHE	Normal
Strategic-P	*Print potential*	DHE	DHE	DHE	Printer
Strategic-A	*Assembly potential*	Cycle-DHE	Cycle-DHE	Cycle-DHE	Assembler

The initial results for running the strategic decision-making algorithm with experimental condition A0 are listed in Table 5.10. The strategic decision-making algorithm outperformed the other decision-making algorithms for the selected optimization metric. However, the base algorithm (HHE configuration) performed the same as the strategic-C algorithm, in terms of collection potential, at time-step 30. Specifically, both the strategic-C algorithm and HHE had a collection potential of approximately 44 (with a standard deviation of 2.98 for HHE and 4.01 for strategic-C) at time-step 30.

Table 5.10. Initial run data of the strategic algorithm for experimental condition A0.

Strategic (Initial)	**Assembly Potential**			**Collection Potential**			**Print Potential**		
	30	**50**	**70**	**30**	**50**	**70**	**30**	**50**	**70**
Strategic-A	34.9	91.1	117.4	40.6	113.7	139.9	5.4	21.3	21.1
Strategic-C	5.8	21.6	22.2	44.3	206.7	238.8	5.8	21.0	21.0
Strategic-P	7.8	23.9	24.2	35.2	120.1	139.9	27.2	94.9	114.2

5.4.2. Hypothesis

The strategic decision-making algorithm is predicted to perform the best in terms of its inputted optimization metric for most experimental conditions. This is inclusive of the results from the variable decision-making algorithm. Likely exceptions to this include the higher task risk experimental conditions such as C7, C8, and C9, due to the high rate of robot failure in these conditions.

A factor that may alter the algorithm's performance is the rate at which the robot system produces robots and the rate at which available resources are depleted. To accommodate conditions that are affected by this, the time-step where the switch from phase one to phase two occurs may need to be adjusted. It is predicted that the phase switch time-step may need to be adjusted upwards to accommodate experimental conditions such as D11 and D12, where the robot system has more resources and therefore takes longer to reach the maximum number of robots. Similarly, in experimental conditions where it takes longer to build robots, such as A10, A12, A14, and A16, the number of robots built in phase one may not have reached a sufficient number to build the robots in phase two. This is predicted to affect the collect, print, and assembly potentials and therefore it will affect all three of the strategic input metrics (strategic-A, strategic-C, and strategic-P).

Second, in the later stages of certain experimental conditions (time-step goal of 70), where available resources constrain the robot system to a lower maximum number of robots, other decision-making algorithms may be able to reach this same maximum. Experimental conditions D1 and D9 (where the initial nonprintable or printable components are reduced) along with experimental conditions B1, B4, B7 (where the gathering rate of resources is decreased) are cases that are predicted to potentially demonstrate this.

5.5. Results

In this section, the results are summarized. Full result tables for the cycle, variable, and strategic decision-making algorithms are provided in Appendix B. In Table 5.11, the results for the base case of A0 (all parameters at default values) are listed for each decision-making algorithm and time-step.

Table 5.11. Results for the decision-making algorithms on experimental condition A0.

A0 Values	**Assembly Potential**			**Print Potential**			**Collection Potential**		
	30	**50**	**70**	**30**	**50**	**70**	**30**	**50**	**70**
CHE	0.99	1.00	1.00	1.00	0.98	1.00	13.77	20.25	29.21
CHO	1.00	1.00	1.00	1.00	1.00	1.00	12.84	18.81	28.24
DHE	22.67	69.88	73.82	18.36	58.22	62.10	41.33	129.48	137.41
DHO	15.70	38.93	82.95	15.66	38.81	82.99	15.78	39.28	83.85
HHE	18.35	63.27	63.79	13.57	53.07	53.81	44.19	164.48	165.53
HHO	13.56	32.19	70.91	13.67	32.19	70.67	23.19	66.46	142.48
Cycle-DHE	20.20	56.92	68.12	18.15	56.99	67.89	38.63	115.44	137.53
Cycle-HHE	10.75	52.61	57.63	10.54	52.35	57.02	31.95	156.38	167.48
Cycle-HHO	11.62	23.71	54.78	11.74	23.74	54.68	25.50	63.14	149.20
VAR-A	22.81	69.71	74.13	18.52	58.58	62.52	41.61	129.52	138.00
VAR-C	18.16	63.80	63.95	13.33	53.50	53.95	43.61	166.27	165.59
VAR-P	22.80	66.97	69.36	18.20	63.02	65.88	41.25	131.39	136.67
STR-A	33.65	91.05	116.90	5.45	21.50	21.35	39.45	113.60	139.85
STR-C	5.95	22.50	21.75	5.95	22.25	21.25	43.95	216.70	234.60
STR-P	7.80	24.60	24.35	25.85	100.40	114.00	33.95	126.10	139.65

An overview of the results of each decision-making algorithm, across an entire classification of experimental conditions, is provided in Tables 5.12 to 5.15. In these tables, the percentage shown is the percent share of the total of each column, in terms of the sum of each decision-making algorithm across the experimental condition classification. Due to the variance of each experimental condition in the experimental condition classification, this percentage may skew to favor higher-performing experimental conditions over low performing experimental conditions. To this end, the values for each decision-making algorithm and experimental condition are listed in Appendix B.

Table 5.12 shows the details for experimental condition classification ‘A’ (28 experimental conditions). Table 5.13 shows the details for experimental condition classification ‘B’ (10 experimental conditions). Table 5.14 shows the details for experimental condition classification

‘C’ (19 experimental conditions). Table 5.15 shows the details for experimental condition classification ‘D’ (12 experimental conditions).

Table 5.12. Percentage of column total on experimental condition classification ‘A’.

Classification ‘A’	**Assembly Potential**			**Print Potential**			**Collection potential**		
	30	**50**	**70**	**30**	**50**	**70**	**30**	**50**	**70**
CHE	0.51%	0.20%	0.15%	0.60%	0.22%	0.17%	3.23%	1.68%	1.82%
CHO	0.51%	0.20%	0.15%	0.60%	0.22%	0.17%	2.99%	1.58%	1.66%
DHE	10.19%	10.16%	9.68%	8.53%	8.51%	7.96%	7.92%	7.44%	7.00%
DHO	7.14%	6.59%	8.58%	8.41%	7.25%	9.47%	3.24%	2.73%	3.55%
HHE	8.62%	8.95%	8.44%	7.01%	7.27%	6.48%	8.92%	9.07%	8.33%
HHO	5.72%	4.90%	6.41%	6.76%	5.39%	7.07%	5.06%	5.03%	6.74%
Cycle-DHE	8.23%	8.75%	8.40%	9.21%	9.76%	9.43%	7.30%	7.33%	7.02%
Cycle-HHE	5.49%	7.28%	6.91%	6.29%	8.02%	7.64%	7.31%	9.14%	8.53%
Cycle-HHO	4.88%	3.62%	4.90%	5.78%	3.99%	5.40%	5.20%	4.84%	6.68%
VAR-A	10.18%	10.21%	9.70%	8.50%	8.52%	7.94%	7.99%	7.52%	7.08%
VAR-C	8.80%	8.60%	7.92%	7.85%	7.99%	7.51%	9.10%	9.32%	8.62%
VAR-P	8.97%	8.91%	8.69%	9.75%	9.50%	9.38%	7.22%	6.95%	6.73%
STR-A	13.98%	13.62%	13.99%	3.29%	4.00%	3.06%	7.61%	7.17%	6.94%
STR-C	2.95%	3.68%	2.81%	3.44%	4.06%	3.10%	9.79%	12.63%	12.24%
STR-P	3.83%	4.34%	3.27%	13.98%	15.28%	15.22%	7.14%	7.57%	7.06%

Table 5.13. Percentage of column total on experimental condition classification 'B'.

Classification 'B'	Assembly Potential			Print Potential			Collection potential		
	30	50	70	30	50	70	30	50	70
CHE	0.47%	0.21%	0.15%	0.54%	0.22%	0.16%	2.95%	1.73%	1.76%
CHO	0.47%	0.21%	0.15%	0.54%	0.23%	0.16%	2.74%	1.65%	1.70%
DHE	9.77%	10.09%	9.55%	9.05%	8.58%	7.81%	8.12%	7.57%	6.94%
DHO	7.28%	6.91%	8.73%	8.41%	7.46%	9.37%	3.35%	2.89%	3.59%
HHE	8.21%	8.57%	7.98%	6.79%	6.79%	6.06%	8.82%	8.78%	8.11%
HHO	6.38%	5.56%	7.39%	7.39%	6.01%	7.94%	4.98%	4.91%	6.66%
Cycle-DHE	8.57%	8.72%	8.33%	9.36%	9.68%	9.29%	7.69%	7.43%	7.01%
Cycle-HHE	4.96%	6.92%	6.93%	5.60%	7.49%	7.39%	6.73%	8.86%	8.73%
Cycle-HHO	5.45%	4.15%	5.84%	6.33%	4.50%	6.26%	5.25%	5.04%	6.98%
VAR-A	9.78%	9.98%	9.40%	9.05%	8.52%	7.70%	8.13%	7.72%	7.02%
VAR-C	8.56%	8.25%	7.59%	7.26%	7.27%	6.86%	8.91%	8.90%	8.57%
VAR-P	9.04%	9.02%	8.51%	9.71%	9.57%	9.21%	7.73%	7.28%	7.01%
STR-A	14.69%	13.16%	13.52%	3.00%	4.27%	3.07%	7.97%	7.19%	6.75%
STR-C	2.76%	3.96%	2.84%	3.14%	4.35%	3.14%	9.40%	12.39%	11.94%
STR-P	3.60%	4.32%	3.10%	13.85%	15.07%	15.57%	7.21%	7.66%	7.24%

Table 5.14. Percentage of column total on experimental condition classification 'C'.

Classification 'C'	Assembly Potential			Print Potential			Collection potential		
	30	50	70	30	50	70	30	50	70
CHE	0.55%	0.19%	0.14%	0.66%	0.21%	0.16%	3.30%	1.50%	1.68%
CHO	0.55%	0.19%	0.14%	0.68%	0.22%	0.17%	3.02%	1.43%	1.60%
DHE	10.07%	10.19%	9.00%	9.46%	9.05%	8.03%	8.42%	7.78%	7.02%
DHO	6.79%	6.01%	9.17%	8.29%	6.56%	9.95%	3.22%	2.54%	3.91%
HHE	8.13%	9.36%	7.74%	6.99%	8.36%	6.94%	8.80%	9.96%	8.39%
HHO	6.07%	4.92%	8.16%	7.43%	5.40%	8.89%	4.73%	4.13%	6.92%
Cycle-DHE	8.43%	8.40%	8.04%	9.18%	9.03%	8.69%	7.50%	7.00%	6.85%
Cycle-HHE	4.74%	7.27%	6.71%	5.54%	7.80%	7.24%	6.42%	8.91%	8.24%
Cycle-HHO	5.14%	3.63%	5.96%	6.30%	4.01%	6.50%	5.13%	4.01%	6.98%
VAR-A	10.05%	10.17%	8.99%	9.40%	9.09%	8.03%	8.39%	7.89%	7.07%
VAR-C	8.19%	9.37%	7.75%	7.00%	8.35%	6.92%	8.84%	10.00%	8.41%
VAR-P	10.05%	9.98%	8.72%	9.40%	9.30%	8.28%	8.39%	7.97%	7.07%
STR-A	15.06%	13.23%	14.14%	2.97%	3.55%	2.64%	8.12%	6.87%	7.03%
STR-C	2.68%	3.38%	2.51%	3.18%	3.57%	2.69%	8.73%	12.47%	11.78%
STR-P	3.50%	3.72%	2.83%	13.54%	15.51%	14.89%	6.98%	7.55%	7.05%

Table 5.15. Percentage of column total on experimental condition classification 'D'.

Classification 'D'	Assembly Potential			Print Potential			Collection potential		
	30	**50**	**70**	**30**	**50**	**70**	**30**	**50**	**70**
CHE	0.45%	0.16%	0.12%	0.53%	0.17%	0.12%	2.83%	1.34%	1.38%
CHO	0.45%	0.16%	0.12%	0.53%	0.17%	0.12%	2.64%	1.25%	1.32%
DHE	10.00%	10.05%	9.24%	9.36%	8.92%	8.44%	8.31%	7.67%	7.27%
DHO	6.81%	6.20%	8.74%	8.05%	6.61%	9.11%	3.15%	2.58%	3.66%
HHE	8.27%	9.15%	7.89%	7.06%	8.28%	7.21%	9.01%	9.91%	8.66%
HHO	5.92%	4.95%	7.53%	7.02%	5.28%	7.82%	4.88%	4.43%	6.59%
Cycle-DHE	8.58%	8.58%	8.42%	9.30%	9.16%	9.05%	7.62%	7.16%	7.17%
Cycle-HHE	4.80%	7.63%	7.12%	5.51%	8.10%	7.59%	6.52%	9.39%	8.72%
Cycle-HHO	5.10%	3.80%	5.87%	6.06%	4.06%	6.10%	5.19%	4.29%	6.54%
VAR-A	10.02%	10.09%	9.27%	9.31%	9.03%	8.46%	8.30%	7.65%	7.23%
VAR-C	8.83%	9.06%	7.83%	7.77%	8.41%	7.30%	9.13%	9.87%	8.64%
VAR-P	9.57%	9.24%	8.76%	9.65%	9.29%	8.92%	8.22%	7.40%	7.16%
STR-A	15.08%	13.37%	13.69%	2.90%	3.67%	2.61%	8.10%	7.00%	6.78%
STR-C	2.63%	3.57%	2.56%	3.05%	3.70%	2.65%	9.04%	12.51%	11.86%
STR-P	3.49%	3.98%	2.85%	13.89%	15.13%	14.49%	7.07%	7.57%	7.02%

5.6. Analysis

In this section, an analysis of the results, as they relate to the previously discussed hypotheses, is presented.

5.6.1. Evaluation of the Cycle Decision-Making Algorithm Hypothesis

The experimental conditions that formed the hypothesis for the cycle decision-making algorithm are provided in Table 5.16, for reference. The accuracy of the hypothesis for the cycle-decision making algorithm is now evaluated.

Table 5.16. Experimental condition reference for cycle decision-making results.

ID	Experimental Condition	Description
A0	*(Default)*	Default values for all parameters.
A3	*BaseCost_Pr + 5*	BaseCost_Pr increased from 2 to 7.
A6	*PrintCost_Pr + 5*	PrintCost_Pr increased from 2 to 7.
A9	*AssembleCost_Pr + 5*	AssembleCost_Pr increased from 2 to 7.
A21	*[All]CostPrintable + 2*	Base-, Print-, and AssembleCost_Pr increased to 4.
A23	*Base & Print Pr + 2*	BaseCost_Pr and PrintCost_Pr increased to 4.
B1	*Print_Efficiency = 0.25*	Print_Efficiency decreased from 1.0 to 0.25.
B7	*Print_Amount = 0.25*	Print_Amount decreased from 1.0 to 0.25.

It was hypothesized that the DHE and HHE configurations would have a higher print potential and lower assembly potential under experimental conditions A3, A21, A23, B1, and B7, as compared to using the base algorithm. Based on the experimentation results, the cycle-DHE and cycle-HHE algorithms outperformed the base algorithm, in terms of print potential, for experimental conditions A3, A21, A23, B1, and B7 (as shown in Table 5.17). This was also the case for experimental conditions A6 and A9, which had the previously discussed ratio discrepancies, when utilizing the base algorithm. These were not predicted to be affected by the use of the cycle algorithm. Furthermore, the assembly potential was higher when the base algorithm was used (as shown in Table 5.18) in all of the listed experimental conditions. Thus, this aspect of the hypothesis was supported by the results.

In broader terms, the cycle-DHE and cycle-HHE algorithm did not outperform the base algorithm in terms of print potential for many experimental conditions during the early stages (up to time-step 30). This is contrary to the prediction that the cycle algorithm would improve the performance of the DHE and HHE configurations in the early stages of operations. At time-steps 50 and 70, however, the cycle-DHE and cycle-HHE algorithms did outperform their base algorithm counterpart in nearly all experimental conditions, in terms of print potential. However, their assembly potential was lower than their base algorithm counterpart for all of the experimental conditions, regardless of time-steps. Thus, the results support the conclusion that the cycle-DHE and cycle-HHE algorithms have an increased or equivalent print potential in later stages, as compared to the base algorithm, for almost all of the tested circumstances; however, they have a reduced assembly potential. This is presumably due to the printer robot type appearing in the build queue more often when using the cycle algorithm with the heterogeneous configurations, as the build list index is retained from where it left off in the previous iteration. Based on the results, this has more of an impact on the later stages. This is contrary to the prediction that it would have more of a significant impact at the early stage (up to time-step 30).

Table 5.17. Print potential of the cycle-DHE and cycle-HHE algorithms.

	Print Potential											
	Time-Step: 30				**Time-Step: 50**				**Time-Step: 70**			
ID	**Cycle**		**Base**		**Cycle**		**Base**		**Cycle**		**Base**	
	DHE	**HHE**	**DHE**	**HHE**	**DHE**	**HHE**	**DHE**	**HHE**	**DHE**	**HHE**	**DHE**	**HHE**
A0	17.9	10.8	18.3	13.5	58.1	50.1	58.0	53.9	67.6	57.5	63.0	53.5
A3	10.2	7.5	5.8	4.8	20.9	12.0	5.8	4.9	34.1	17.2	5.7	4.7
A6	11.8	10.5	10.6	10.5	28.5	28.0	19.2	10.7	41.4	34.7	27.3	9.8
A9	15.8	10.8	11.1	11.9	41.9	37.1	25.5	28.3	59.1	45.4	42.9	40.1
A21	11.1	9.3	6.8	6.9	24.2	15.0	6.7	6.9	37.7	26.4	6.6	6.4
A23	11.4	9.3	8.8	7.9	27.9	20.8	8.8	7.7	40.9	31.8	8.2	7.4
B1	14.0	10.6	12.2	10.8	21.9	16.7	12.1	10.4	25.9	19.7	11.7	10.4
B7	16.3	9.9	12.6	10.9	26.0	15.9	12.3	10.4	43.1	20.8	12.3	10.0

Table 5.18. Assembly potential of the cycle-DHE and cycle-HHE algorithms.

	Assembly Potential											
	Time-Step: 30				Time-Step: 50				Time-Step: 70			
ID	Cycle		Base		Cycle		Base		Cycle		Base	
	DHE	HHE	DHE	HHE	DHE	HHE	DHE	HHE	DHE	HHE	DHE	HHE
A0	19.7	10.8	22.8	18.5	58.4	52.4	68.8	64.2	68.9	57.6	74.7	64.7
A3	10.0	6.6	13.4	10.6	18.5	10.2	21.9	15.4	29.3	18.0	34.1	21.5
A6	15.6	10.8	20.5	17.1	40.1	33.3	52.5	53.2	58.6	44.3	81.9	84.1
A9	12.3	10.6	15.1	13.5	28.9	29.5	30.0	30.0	38.7	36.9	44.7	42.4
A21	11.4	9.7	15.5	12.8	22.0	17.3	26.9	22.6	35.0	29.4	41.0	34.9
A23	13.3	11.0	19.3	15.7	28.2	21.6	40.4	32.3	42.7	32.6	67.9	51.7
B1	14.8	10.7	16.2	15.0	18.3	13.6	26.2	17.8	24.6	19.4	39.6	24.3
B7	15.2	10.1	18.3	15.4	22.8	14.4	27.8	18.4	36.0	20.8	41.6	24.1

For the cycle-HHO algorithm, there was a marginal increase in the collection potential at time-step 30, as compared to the base algorithm HHO (as shown in Table 5.19). However, the cycle-HHO algorithm underperformed the base algorithm HHO by a small amount in later timesteps. Therefore, the predicted better performance of the cycle-HHO algorithm, in terms of collection potential, for these cases, is largely unsupported by the results.

The cycle-HHO algorithm had a slightly lower assembly and print potential for experimental conditions A6, A9, A21, A23, B1, and B7 (as shown in Table 5.20). A more significantly inferior performance was shown under experimental condition A3, where the cycle-HHO algorithm appears to have stalled the production of replicator robots. However, this stall in replicator robot production is shown in the other listed experimental conditions as well – it just did not occur for experimental condition A3 when using the base algorithm.

Due to the Cycle-HHO algorithm underperforming its base algorithm counterpart in most experimental conditions, it is not projected to be a strong option for use in applications similar to most of the tested circumstances.

Table 5.19. Collection potential for the cycle-HHO algorithm.

Collection Potential						
Time-Step:	**30**		**50**		**70**	
ID	**Cycle HHO**	**Base HHO**	**Cycle HHO**	**Base HHO**	**Cycle HHO**	**Base HHO**
A0	24.4	23.4	66.9	67.3	144.8	142.7
A3	15.1	13.1	24.7	26.8	34.1	44.8
A6	25.5	26.3	59.1	62.2	95.7	106.0
A9	24.8	25.6	61.5	63.2	98.1	105.5
A21	18.8	19.9	35.5	38.0	59.9	65.4
A23	20.4	21.0	39.9	44.1	63.5	71.1
B1	23.7	21.5	43.2	44.9	64.8	73.4
B7	23.3	21.3	41.6	44.1	66.4	73.8

Table 5.20. Assembly potential and print potential for the cycle-HHO algorithm.

	Assembly Potential						**Print Potential**					
	30		**50**		**70**		**30**		**50**		**70**	
ID	**Cycle HHO**	**Base HHO**	**Cycle HHO**	**Base HHO**	**Cycle HHO**	**Base HHO**	**Cycle HHO**	**Base HHO**	**Cycle HHO**	**Base HHO**	**Cycle HHO**	**Base HHO**
A0	11.3	13.5	24.1	32.2	51.6	70.1	11.4	13.7	23.9	31.8	51.8	70.4
A3	6.0	9.7	5.9	16.7	5.9	31.3	6.0	9.7	5.8	16.7	5.7	31.0
A6	8.0	8.8	7.7	8.8	7.7	8.5	8.0	8.8	7.9	9.0	7.5	8.8
A9	7.7	8.8	7.9	8.7	7.7	8.3	7.9	8.8	7.9	8.9	7.7	8.7
A21	6.8	7.8	6.8	7.7	6.9	7.9	6.8	7.9	6.9	7.8	6.8	7.7
A23	7.7	8.7	7.5	8.9	7.8	8.7	7.8	8.8	7.6	8.8	7.8	8.4
B1	11.4	13.7	12.6	14.7	12.4	14.5	11.6	13.7	12.3	14.6	12.0	14.1
B7	11.3	13.7	12.6	14.3	12.5	14.5	11.4	13.8	12.3	14.3	12.3	14.4

5.6.2. Evaluation of the Variable Decision-Making Algorithm Hypothesis

In this subsection, the hypothesis regarding the variable decision-making algorithm is evaluated and compared with the results obtained.

The variable-P algorithm was within standard deviation of the highest print potential for most experimental conditions at time-steps 40 and 50. It also performed the best for approximately 35% of all experimental conditions and the second best in an additional 35% of all experimental conditions, for both time-steps 40 and 50. Considering standard deviation, at time-step 40 it underperformed the base-DHE algorithm for experimental condition A16, and Cycle-DHE for experimental conditions A6 and A13 (as shown in Table 5.21). At time-step 50, it underperformed as compared to the Cycle-DHE algorithm for experimental conditions A6, A13, A16, and A23 as well as compared to base-DHO, for experimental condition D9. Thus, the prediction that the variable-P algorithm would perform the best for most experimental conditions at time-steps 40 and 50 is supported by the results.

At time-step 70, the variable-P algorithm was hypothesized to underperform for experimental conditions A11, A13, A15, A17, A18, A19, A25, A26, A27, A28, B3, B6, B9, D2, D5, D6, and D9 due to reaching the maximum number of robots as limited by resource constraints. Based on the results, it underperformed under all of the experimental conditions that were hypothesized, as well as for A0, A7, B5, D8, D7, D4, and D10. This supports the hypothesis that it would reach a suboptimal maximum, in terms of print potential, for the experimental conditions where the heterogeneous approach underperformed the homogeneous approach in Chapter 4.

Table 5.21. Select results for the variable-P algorithm at time-steps 40 and 50.

VAR-P	**Print Potential**				
ID	**Highest**	**Score**	**VAR-P**	**Diff**	**Std Dev**
(40)	**Time-Step: 40**				
A6	Cycle-DHE	18.9	14.9	*-4.0*	1.2
A13	Cycle-DHE	41.5	37.2	*-4.3*	2.9
A16	DHE	13.4	9.5	*-3.9*	1.0
(50)	**Time-Step: 50**				
A6	Cycle-DHE	28.6	18.3	*-10.3*	2.3
A13	Cycle-DHE	69.4	57.9	*-11.5*	4.7
A16	Cycle-DHE	24.6	18.0	*-6.6*	2.0
A23	Cycle-DHE	26.8	21.7	*-5.1*	2.7
D9	DHO	39.0	35.0	*-4.0*	2.0

The variable-A algorithm was within the standard deviation of the highest performing algorithm, in terms assembly potential, for all experimental conditions at time-step 50. At time-step 40, it was within the standard deviation of the best performing algorithm for all experimental conditions except B5. For experimental condition B5, it underperformed, as compared to the base-DHE configuration (as shown in Table 5.22). Thus, the prediction that the variable-A algorithm would outperform the base and cycle algorithms for most experimental conditions at time-steps 40 and 50, is supported by the results.

At time-step 70, it underperformed under all of the experimental conditions hypothesized (the same conditions hypothesized for the variable-P algorithm), as well as under experimental conditions A7, B5, D7, and D8. This supports the hypothesis that it would reach a suboptimal maximum, in terms of assembly potential, for the experimental conditions where the heterogeneous approach underperformed the homogeneous approach in Chapter 4.

Table 5.22. Select results for the variable-A algorithm at time-step 40.

(40)	**Assembly Potential**				
ID	**Highest**	**Score**	**VAR-A**	**Diff**	**Std Dev**
B5	DHE	33.4	30.3	*-3.2*	2.2

The variable-C algorithm did not perform nearly as well as predicted. It had some success in the earlier stages, except for under experimental conditions A10, A26, A27, B8 and D2, where it wasn't within the standard deviation range from the best performing algorithm at time-step 40 (as shown in Table 5.23). For time-steps 50 and 70, it underperformed across several experimental conditions. Therefore, the prediction that the variable-C algorithm would outperform other algorithms for all experimental conditions is unsupported by the results.

Table 5.23. Select results for the variable-C algorithm at time-step 40.

(40)	**Collection Potential**				
ID	**Highest**	**Score**	**VAR-C**	**Diff**	**Std Dev**
A10	Variable-A	31.2	27.2	*-4.0*	2.0
A26	HHE	105.5	94.5	*-11.0*	7.0
A27	HHE	103.7	90.3	*-13.4*	7.3
B8	Cycle-HHE	58.5	51.0	*-7.5*	2.9
D2	HHE	112.1	104.8	*-7.3*	6.2

The variable-P and variable-A algorithms did not perform better than the base and cycle algorithms on the high-risk experimental conditions of C7, C8, and C9. The variable-C algorithm performed the best under experimental conditions C8 and C9 at time-steps 40 and 50, although the difference between them was less than half of the standard deviation (the standard deviation on those experimental conditions is notably quite high). Therefore, this portion of the hypothesis is weakly supported with regards to the variable-C algorithm being able to better cope with high task risk levels. However, this result is inconclusive.

5.6.3. Evaluation of the Strategic Decision-Making Algorithm Hypothesis

Due to their high task risk levels, experimental conditions C7, C8, and C9 were predicted to be an exception to the strategic algorithm outperforming the other algorithms, in terms of the selected optimization metric. This prediction is supported by the results. In certain cases, the strategic decision-making algorithm outperformed other algorithms under one or more of these experimental conditions. However, the strategic algorithm's observed performance over the other algorithms was limited for these conditions (even at time-step 70). This data, thus, supports the prediction that these high-risk experimental conditions are an exception. However, this is identified as an exception due the to experimentation being inconclusive and not an indication of consistent better or worse performance. Because of this, the discussions of the results herein generally excludes experimental conditions C7, C8, and C9, due to their inconclusiveness.

The strategic-A algorithm was the highest performing algorithm in terms of assembly potential for most experimental conditions and time-steps (shown in Table 5.24). At time-step 30, this algorithm marginally underperformed (but it was within the standard deviation range) the variable-A algorithm for experimental condition A3. It also underperformed the base-DHE algorithm for experimental condition A14. Furthermore, it marginally outperformed the other algorithms under experimental conditions A10, A16, A21, and C19. It substantially (12%-42%) outperformed for the other experimental conditions.

At time-step 50, the Strategic-A algorithm marginally underperformed (but was within the standard deviation range of) the base-DHE algorithm under experimental condition B1. It marginally outperformed under experimental conditions A12, A14, A23, B4, B7, C19, and D9, and substantially outperformed (by 14%-35%) under the other experimental conditions. At time-step 70, it marginally underperformed the base-DHE algorithm for experimental condition B1. However, it marginally outperformed for experimental conditions A16, A17, B7, C10, D8, and D9, and substantially outperformed (15%-43%) for the other experimental conditions.

Table 5.24. Select results for the strategic algorithm.

	Assembly Potential (Strategic-A)			Collection Potential (Strategic-C)			Print Potential (Strategic-P)		
ID	***30***	***50***	***70***	***30***	***50***	***70***	***30***	***50***	***70***
A0	33.7	91.1	117.0	44.0	216.7	234.6	25.9	100.5	114.0
A3	13.5	26.9	48.9	24.4	46.9	78.0	13.8	24.1	53.5
A6	28.7	68.8	123.5	44.0	145.1	243.2	17.1	41.1	58.3
A9	16.0	37.2	56.6	44.4	145.2	246.2	26.3	59.7	111.1
A10	8.0	42.9	95.5	15.0	44.5	138.5	7.4	37.9	96.9
A12	32.0	65.0	117.7	36.3	150.9	235.9	13.5	61.7	116.2
A14	11.2	45.5	90.6	27.2	111.9	217.2	15.3	62.3	120.6
A16	8.0	41.4	75.1	25.6	84.8	163.2	7.7	38.3	61.3
A17	35.3	67.2	67.5	43.2	115.0	115.4	26.5	66.2	66.6
A21	15.8	33.0	59.8	37.2	77.5	127.6	15.9	30.0	60.1
A23	21.3	42.6	79.9	37.7	81.4	133.6	17.9	36.0	61.2
A25	45.6	109.7	109.3	44.2	216.3	222.7	27.1	100.2	99.4
A26	41.7	108.6	112.0	45.1	227.0	232.0	26.3	107.1	110.3
A27	36.5	97.4	115.2	42.4	223.3	231.7	27.4	109.3	111.1
A28	45.0	112.4	111.5	43.0	221.9	221.0	26.3	104.3	105.7
B1	22.6	24.3	37.1	42.5	59.0	80.3	21.4	26.9	37.8
B4	32.7	41.4	82.2	43.6	114.4	221.5	26.1	39.9	85.3
B7	23.6	29.5	47.6	44.0	66.8	106.2	23.8	29.5	78.8
B9	37.2	112.3	117.6	43.4	222.2	234.0	25.9	110.2	113.1
D1	23.1	76.6	123.6	44.2	176.7	248.6	23.5	78.4	120.8
D2	44.1	111.4	113.0	42.0	226.8	228.0	26.4	104.1	102.9
D5	39.3	96.2	113.1	42.2	230.2	234.8	27.6	108.1	113.3
D8	35.4	88.8	88.2	43.6	194.8	167.6	26.8	99.6	97.5
D9	34.7	42.7	44.0	43.8	93.1	92.6	26.5	45.0	43.7

The strategic-C algorithm had the highest collection potential for most experimental conditions at time-steps 50 and 70. At time-step 30, it performed comparatively well, as compared to the other algorithms, but it was not the highest, under most experimental conditions. At time-step 50, the strategic-C algorithm notably underperformed the base-DHE algorithm for experimental condition A10 (it was not even within the standard deviation range). It also marginally underperformed the base-HHE algorithm under experimental condition C4. However, it outperformed under the other experimental conditions by a substantial margin (10%-33%). At time-step 70, it notably underperformed the base-HHE algorithm under experimental condition A10. It marginally outperformed for experimental conditions A16, B1, and C19, and outperformed under the other experimental conditions by 10%-47%.

The strategic-P algorithm had the highest print potential for most experimental conditions and time-steps. In fact, it was the highest performing for all experimental conditions at time-steps 50 and 70 (excluding C7, C8, and C9, as noted previously). At time-step 30, it marginally underperformed the base-DHO algorithm under experimental condition A10. It marginally outperformed under experimental condition A3 and notably outperformed under the other experimental conditions by 18%-45%.

At time-step 50, the strategic-P algorithm was the highest for all experimental conditions. It marginally outperformed for experimental conditions A3, B7, and D9. However, it outperformed under the other experimental conditions by 21%-44%. At time-step 70, it outperformed for all experimental conditions. It marginally outperformed for experimental conditions A16, A17, and D9, and outperformed under the other experimental conditions by 12%-44%.

Based on this data, the hypothesis that the strategic decision-making algorithm would outperform the other algorithms in this experiment, in terms of its input optimization metric, for most experimental conditions is supported. However, under experimental conditions A10, A12, A14, and A16 the strategic algorithm underperformed, in terms of its input metric, in many cases. Experimental conditions B1, B4, B7, and B9 showed a similar effect. These experimental conditions primarily affected its performance at time-steps 50 and 70.

Experimental conditions D11 and D12 did not show a significant impact on the performance; therefore, this portion of the hypothesis is not supported by the data. Furthermore, experimental condition D9 was close in certain cases, primarily for assembly and print potentials.

Experimental condition D1 did not cause a reduction in the performance of the strategic algorithm significantly.

5.6.3.1. Strategic-A Improvements

The Strategic-A algorithm underperformed under experimental condition B1 at time-steps 50 and 70 (depicted in Figure 5.3). This experimental condition lowers the *Print_Efficiency* parameter, which makes the conversion of raw materials to printable components less efficient. Thus, the problem presented to the algorithm had two components. First, it needed to increase the number of print-capable robots in order to offset this less efficient conversion. Second, it needed an increase in collection potential early in the simulation in order to provide the materials for the larger number of print-capable robots required.

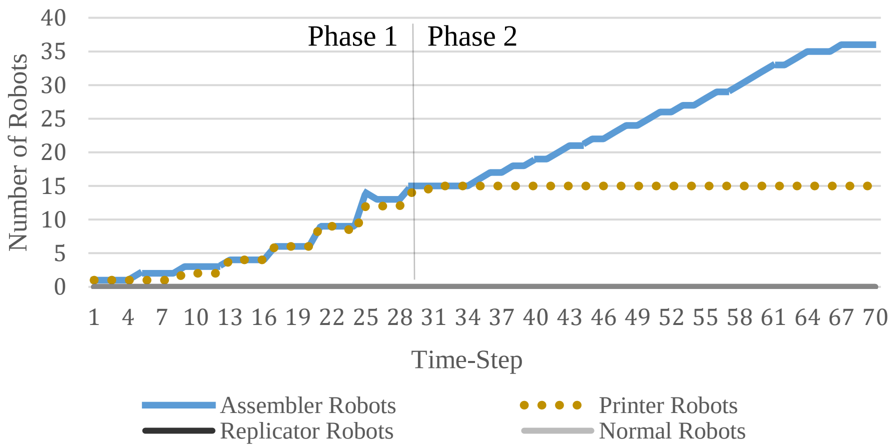


Figure 5.3. The STR-A algorithm using cycle-DHE for phase 1 on experimental condition B1.

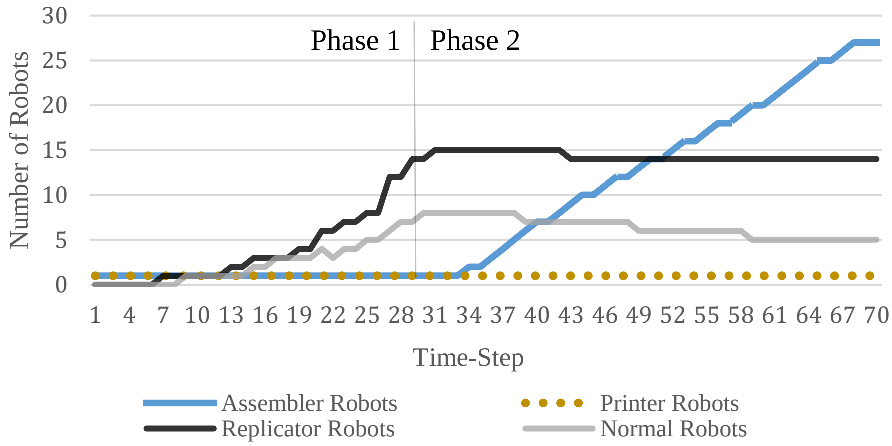


Figure 5.4. The STR-A algorithm using *R-R-N* for phase 1 on experimental condition B1.

A solution was found which involved using a custom build list that consisted of two replicator robots followed by a normal robot *(R-R-N)*. Based on the output of trial simulation runs, this produced acceptable performance due to the replicator robots being primarily tasked with assembling new robots during phase one, and then being tasked with printing components during phase two (depicted in Figure 5.4). The assembler type robots are prioritized over the replicator type robots by the algorithm for assembling new robots, since they don't have the capability to print components. With this change in phase one build list, the strategic-A algorithm outperformed in terms of assembly potential for experimental condition B1 at time-steps 50 and 70 (as shown in Table 5.25).

Table 5.25. Improvement of the strategic-A algorithm phase 1 on experimental condition B1.

Strategic-A Condition: B1	Assembly Potential	Assembly Potential Std Dev	Previous Best	Previous Best Score
Time-Step 50	28.67	*2.04*	DHE	25.73
Time-Step 70	42.22	*3.18*	DHE	38.72

5.6.3.2. Strategic-P Improvements

Based on its superior performance during the experimental runs (in terms of its optimization metric), the strategic-P algorithm was not further modified. The results produced indicate a high level of performance and robustness across many different experimental conditions using its initial phase one build list.

5.6.3.3. Strategic-C Improvements

There were several changes made to the time-step 30 phase one build list for the strategic-C algorithm, due to its relatively low initial performance. First, phase one was changed to use the HHE configuration (from the cycle-DHE configuration) for all cases at time-step 30. Based on the results of running all experimental conditions with this change to the algorithm's early stage, it had marginal to significant improvements on many experimental conditions. However, with these changes it now performed worse under experimental conditions A2, A3, A11, A21, A23, and A24, when using the HHE configuration as the phase 1, as compared to when using the cycle-DHE configuration for phase 1. The algorithm was switched back to use cycle-DHE for these experimental conditions.

Even with the change to the operations of the algorithm by using the HHE configuration during phase 1, it still underperformed under experimental conditions A25, A26, A27, and A28, in terms of collection potential. Interestingly, the strategic-A algorithm outperformed the strategic-C algorithm under experimental conditions A25, A26, A28, and D2, in terms of collection potential. This was unexpected as the strategic-A algorithm was designed for maximizing assembly potential. Experimental conditions A25, A26 and A28 have reduced costs of printable components of the base and/or assemble capability. Thus, in these cases, where a system could build a high number of robots with the initial available resources, an increase in assembly capable robots allows more robots to be built and not be bottlenecked by lack of resources. The phase one operations approach of these experimental conditions was changed to only produce assembler type robots, which resulted in the strategic-C algorithm outperforming for experimental conditions A25, A28, and D2 at time-step 30 by a wide margin (listed in Table 5.26).

Table 5.26. Improvements to the phase 1 of the strategic-C algorithm at time-step 30.

Time-Step: 30	Phase 1	Collection Potential	Collection Potential Std Dev	Previous Highest	Previous Score
A25	Assembler	97.07	*8.52*	Strategic-A	51.45
A26	DHE	55.60	*3.06*	Strategic-A	47.25
A27	DHE	54.56	*6.17*	Variable-A	46.31
A28	Assembler	85.80	*6.43*	Strategic-A	50.65
D2	Assembler	88.09	*5.19*	Strategic-A	49.90
D5	DHE	55.51	*3.41*	Variable-C	48.25

While prioritizing assembly potential during the early period of the simulation benefited the strategic-C algorithm under experimental condition A26, it was not sufficient. Experimental condition A26, where the printable component cost of the assembly capable robots is decreased, still required a certain amount of print capable robots in order to augment the printable component supply. In this regard, using the cycle-DHE or HHE configurations would produce too many print capable robots, and using only assembler robots (which benefited performance for experimental conditions A25 and A28) resulted in running out of printable components too early. Thus, the DHE configuration was tried for phase 1, as this configuration resulted in a higher assembly potential in the experimental runs. This sufficiently increased its assembly potential for the algorithm to exhibit sufficient performance under experimental condition A26 at time-step 30.

Under experimental condition A27, where the printable component cost of the print capability is decreased, the configuration used was also changed to DHE (which improved it enough to cause it to outperform other algorithms). This change was also applied for use under experimental condition D5 (increased initial raw printing materials) and it produced improved results. Experimental condition D5 was identified as possibly benefiting from this change as well, due to the variable-C algorithm having equivalent performance to the variable-A algorithm under this condition. This led to the realization that increased assembly potential was beneficial for this experimental condition.

Table 5.27. The STR-C algorithm on experimental condition A10.

Algorithm	**Phase 1**	***Switch Phase Step***	**Collection Potential**	***Collection Potential Std Dev***
	Time-Step: 30			
Base-HHE	-	-	15.55	*0.89*
STR-C	Cycle-DHE	*15*	15.23	*1.16*
STR-C	*(A-A-P)*	*15*	21.58	*1.21*
	Time-Step: 50			
Base-DHE	-	-	54.78	*3.45*
STR-C	Cycle-DHE	*30*	45.72	*5.78*
STR-C	*(A-A-P)*	*30*	82.02	*8.03*
	Time-Step: 70			
Base-HHE	-	-	161.32	*9.53*
STR-C	Cycle-DHE	*30*	85.27	*11.10*
STR-C	*(A-A-P)*	*30*	149.41	*16.27*
STR-C	*(A-A-P)*	*40*	179.75	*16.75*

Finally, the strategic-C algorithm was underperforming for experimental condition A10 at time-steps 50 and 70. Many changes to its phase 1 operating configuration were attempted to improve this. Experimental condition A10 involves increasing the *BaseCost_Time* parameter, and more assemble-capable robots were needed to be built in phase 1 in order to compensate for the slower build rate. Thus, a customized build list of two assembler robots and one printer robot *(A-A-P)* was used. This resulted in an increase in assembly-capable robots; therefore, more robots could be built in parallel (depicted in Figure 5.5). This increased the performance of the algorithm and achieved the highest level of collection potential for experimental condition A10 at time-step

50. However, even with this change, the strategic-C algorithm still underperformed the base-HHE algorithm at time-step 70 (as shown in Table 5.27).

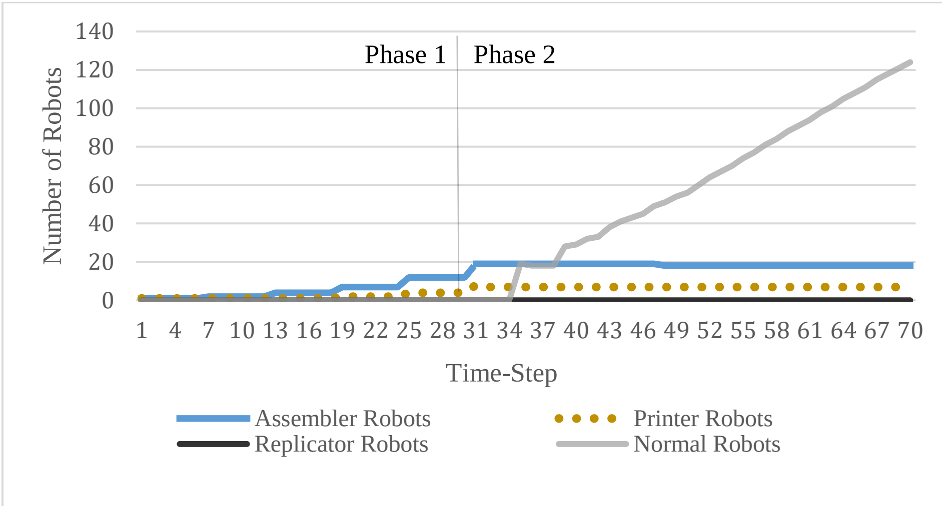


Figure 5.5. The STR-C algorithm using *A-A-P* for phase 1 on experimental condition A10.

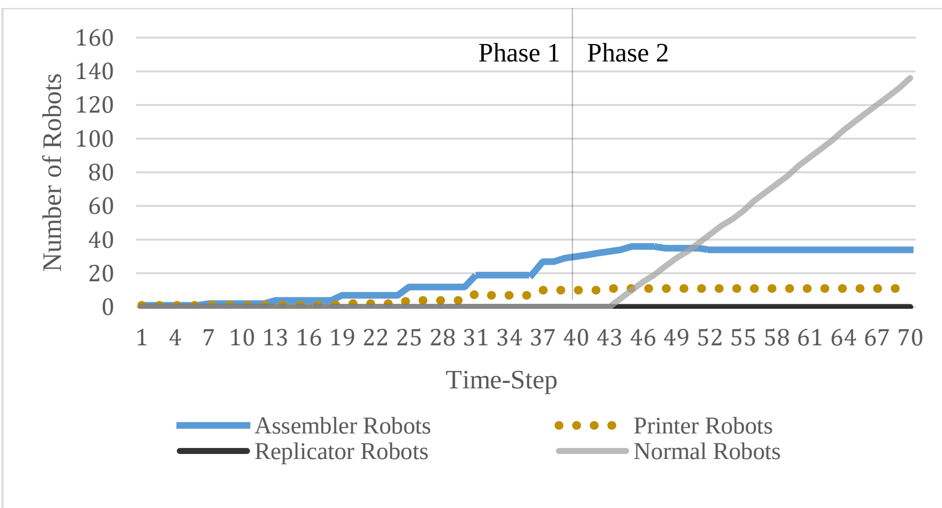


Figure 5.6. The STR-C algorithm using *A-A-P* for phase 1, with the phase switch at step 40.

Due to the increase of the *BaseCost_Time* parameter in experimental condition A10, the strategic-C algorithm required even more assemble-capable robots to assemble in parallel in order to reach a higher collection potential at time-step 70. It was hypothesized that the switch phase time-step may need to be adjusted upwards for experimental conditions A10, A12, A14, A16, D11, and D12. Thus, the time-step where phase one switches to phase two was changed to 40 (up from 30) for experimental condition A10 (as depicted in Figure 5.6). This adjustment increased the performance of the strategic-C algorithm and achieved the highest level of collection potential for experimental condition A10 at time-step 70 (as shown in Table 5.27).

5.7. Summary

This chapter has considered decision-making algorithm performance under multiple experimental conditions that might be present in various real-world operating environments. The type of decision-making algorithm used for a self-replicating robot system has a significant impact on system performance. In this chapter, three decision-making algorithms were proposed and tested. These algorithms were the cycle, variable, and strategic decision-making algorithms.

The cycle decision-making algorithm demonstrated improved print potential in the later time-steps for the DHE and HHE configurations in comparison to the base algorithm. However, it also had decreased assembly potential under most experimental conditions and at most time-steps (due to differences in build priorities).

The results from the base and cycle decision-making algorithms were used to construct the variable algorithm. This algorithm takes an input metric of assembly, print, or collection potential and attempts to maximize for this input by selecting the highest scoring system configuration and decision-making combination from the base and cycle algorithms. However, it did not perform well in terms of collection potential and had problems under experimental conditions where the maximum number of robots was reached early in the simulation.

Based on the variable algorithm's lack of sufficient improvements over the base and cycle decision-making algorithms, the strategic decision-making algorithm was devised. The variable algorithm can be characterized as a greedy algorithm, where the best performing option was selected without regard to looking ahead at subsequent steps. In contrast, the strategic algorithm has a time-step goal in addition to an inputted optimization metric. Thus, its approach is to build a core number of robots in an initial phase, and then focus entirely on maximizing the input metric in a subsequent phase. It performed far better than the other algorithms, in terms of its inputted optimization metric, for almost all experimental conditions and time-steps. Adapting the initial phase 1 build list was shown to be effective in improving its performance under experimental conditions where the strategic algorithm did not initially outperform the other algorithms. The time-step at which to switch from phase 1 to phase 2 was also changed, in one instance, based on the number of time-steps required for building robot-types. Overall, the results presented show the strategic algorithms to be a strong option for maximizing a particular capability metric. Although, it often maximizes the input metric at the cost of other metrics. Given this, it is not suitable for use cases that require a more even distribution of metrics.

6. CYBERSECURITY[6]

Self-replicating robot systems may pose security concerns based on their capabilities and the boundaries of their operations. A potential solution is to strictly set parameters within their software that limit certain behaviors or capabilities. However, this prospective solution wouldn't be effective if the robot system, or a portion of the robot system, was infected through a cybersecurity vulnerability. The potential for the compromise of the safeguards impairs the integrity of this approach. As there is no absolute solution to this problem, the goal becomes to minimize the likelihood of its occurrence. This issue could potentially be mitigated through the use of an anomaly detection system.

An intrusion detection system identifies unusual activity that deviates from the expected activity and indicates the potential compromise of the monitored systems. Detecting deviations from expected behavior using anomaly detection, as opposed to using signature-based detection, allows new variants of malicious activity to be caught by the intrusion detection system [109][110]. However, this approach has associated disadvantages as well. For instance, these detection methods are likely to raise false alarms, such that unusual but legitimate use may be flagged as anomalous [111][112]. To maximize effectiveness, the challenge is to develop an accurate and complete model of legitimate behavior.

In this chapter, the cyber security considerations for self-replicating robot systems are discussed. An experiment is conducted to evaluate the efficacy of a potential anomaly detection

---

[6] This chapter is derived from: A. Jones and J. Straub, "Cybersecurity Considerations for Self-Perpetuating Robot Systems," *(submitted to) J. Comput. Sci. Technol.*, 2020.

system for self-replicating robot systems. The results of this experiment are used to derive a potential anomaly detection system for a self-replicating robot system.

6.1. Robot Cybersecurity

In this section, background information and prior work pertaining to robot cybersecurity is reviewed. Safeguarding autonomous robots from cybersecurity threats is particularly important, as compromised robots can directly and catastrophically impact their surroundings [112]. For example, Bonaci and Chizeck [113] discussed security concerns for remotely operated robots that are used for rescue and recovery in natural disasters and man-made catastrophes, including battlefield environments.

Robotic communication channels are one of the most important security aspects of the system to be secured, due to information that is critical to the mission being sent through these wireless communication channels [114]. To provide security for this, Schumann, Moosbrugger, and Rozier [115] proposed the R2U2 approach, which was designed to continuously monitor inputs from the ground control station, sensor readings, actuator outputs, and flight software status. By monitoring these inputs and performing statistical reasoning, attack patterns and post-attack discrepancies in the robotic behavior could be detected [115].

Robotic systems that operate autonomously and self-organize may be susceptible to special forms of attacks. Higgins, Tomlinson, and Martin [116] discussed how systems using swarm intelligence could be tampered with by an intruder. For example, a rogue robotic agent could be introduced into the robot system and cause unexpected behavior. A proposed mitigation strategy for this is to develop new forms of authentication for robot systems such as the use of visual sensing and physical data exchange [116]. Preventing robots from becoming infected through cybersecurity vulnerabilities, is the focus of the experiment in this chapter.

6.2. Methodology

In this section, the methodology used in this chapter's experiment is discussed. The experiment was conducted to analyze the impact that the system configuration of the self-replicating system would have on cybersecurity related vulnerabilities. The experiment is a modification of the simulation system presented the previous chapters. In the experiment, a certain robot may be infected at a specified time-step.

The particular characteristic being studied is how the system configuration of the self-replicating robot system affects its survival. Survival depends on how many robots are infected and what types they are. For example, in Figure 6.1, if the sole replicator robot is infected (after building the non-replicators) then the system becomes a non-replicating robot system, as the infected robot would need to be deactivated (or would otherwise be unavailable for system operations). Examples where replicating robots are infected are evaluated to determine the impact on the overall system.

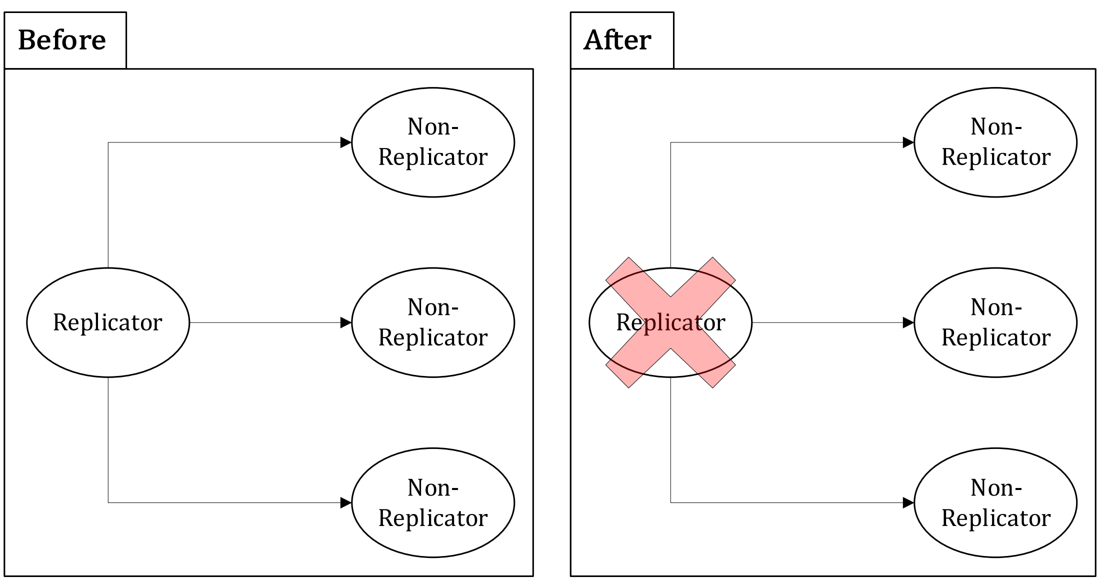


Figure 6.1. Diagram of a centralized self-replicating robot system, before and after infection.

While the impact of different types of malware on system operations will vary significantly, some assumptions were made for experiments presented in this chapter. An infected robot that assembles a new robot will cause the newly constructed robot to also be infected. An infected robot tasked with gathering resources will take resources out of the environment but won't add them to the robot system's stockpile (i.e., it destroys or hides them). Similarly, an infected robot that is tasked with printing components will use printing materials but will not add the printed components to the robot system's stockpile (i.e., it wastes them by producing unusable components).

6.2.1. Experimental Conditions

For this experiment, there are three experimental conditions. These are as follows:

- **Random (T1)**: The robot that will be infected is randomly chosen from all currently operational robots in the system. This is meant to simulate a case where an attacker searches for a robot that has a cyber security vulnerability that could be exploited.
- **Targeted (T2)**: The robot that will be infected is randomly chosen from all assembly-capable robots that are currently operational in the robot system. This is meant to simulate a case where an attacker specifically targets robots that have the assembly capability.
- **Physically Separated (T3)**: The robot that will be infected is randomly chosen from all robots in the robot system that are currently collecting resources. This is meant to simulate a case where an attacker targets a robot that is currently away from the main part of the robot system. This simulates a case where proximity or physical access would be needed (i.e., if the attacker would need to physically access the hardware to compromise it).

For these experiments, the *Num_Steps* parameter of the simulation is set to 70. All other parameters are set at their default values (listed in Chapter 4). A new parameter *step_infected* is used and is varied. This parameter determines which time-step a robot gets infected at. Experiments are first run with a value of 10, this is then changed to 20 and then finally changed to 40. This parameter is varied only for experimental conditions T1 and T2. It is not varied in the T3 condition because the T3 condition has a circumstance that the other two experimental conditions do not: a robot is not guaranteed to be collecting resources at the specified time-step. Thus, for experimental condition T3, the *step_infected* parameter is the first possible time-step where a robot could get infected (i.e., when robots are assigned the collect task).

6.2.2. Hypotheses

The hypotheses of the experiment are presented in this subsection. Specifically, it is predicted that the effect of a robot becoming infected will be reduced proportionately to how late the infection occurs in the simulation run. For the different replication approaches, the hypotheses are as follows:

- **Centralized**: The centralized configurations of CHE and CHO are expected to be significantly affected by experimental condition T2, and only marginally affected by experimental conditions T1 and T3. This is predicted because the centralized configurations are restricted to the initial amount of assembly capable robots. Therefore, these configurations are more likely to have a normal robot infected by a random target attack (experimental condition T1) or by an attack targeted at a remote robot that is collecting resources (experimental condition T3). In contrast, under experimental condition T2, the sole assemble capable robot would be targeted and therefore any subsequently assembled robots would be infected. The number of robots infected, and the amount of raw printing materials lost due to infected robots is estimated to be high in experimental condition T2, but minimal for experimental conditions T1 and T3. Furthermore, the number of fabricated components lost is expected to be higher for the CHO configuration under experimental condition T2, as the CHE configuration has the print and assemble tasks separated amongst more than one robot type (and the printer robot wouldn't be targeted in this case).

- **Decentralized**: The decentralized configurations of DHE and DHO are expected to be impacted by experimental conditions T1, T2, and T3 more than the HHE and HHO configurations. This is predicted because the DHE and DHO configurations do not produce normal robots which could potentially be infected (instead of an assembly capable robot). However, the DHE configuration is predicted to be less affected by experimental conditions T1 and T3, in comparison to the DHO configuration. This is predicted because the DHE configuration has separate print capable robots. This reduces the chance of an assembly capable robot being infected by potentially fifty percent (or the current assemble ratio value) for experimental condition T1 and produces a similar reduction for experimental condition T3 (although this would depend on how the collect task is delegated). Due to this, the DHE configuration is predicted to have a lower percentage of infected robots, on average, for experimental conditions T1 and T3. In contrast, the DHO configuration is expected to perform consistently across the experimental conditions.
- **Hierarchical**: The hierarchical configurations of HHE and HHO are predicted to be affected by experimental condition T2 more significantly than experimental conditions T1 and T3. This is predicted because the hierarchical configurations produce normal

robots, which could potentially be infected (instead of an assembly capable robot). This is especially expected for experimental condition T3, due to the normal robots being a higher proportion of robots collecting resources from the environment. The HHE configuration is predicted to be less affected than the HHO configuration, on average, for experimental conditions T1 and T3, due to the split of the assembly and print capabilities amongst more than one robot type. The HHE configuration tended to produce robots the fastest, in the results presented in previous chapters. Because of this, the HHE configuration may have a higher number of infected robots than other system configurations, but it may have a lower percentage of the total number of robots becoming infected.

6.3. Results

In this section, the results of the experiment are presented. The reported metrics and attributes are as follows:

- **Num Robots**: the number of robots currently in the robot system.
- **Total Collected**: the total amount of raw printing materials collected via the collect task type.
- **Total Printed**: the total amount of printable components fabricated from raw printing materials via the print task type.
- **Num Infected**: the number of infected robots in the robot system.
- **Num Infected Std Dev**: the standard deviation of the number of infected robots (from the 100 runs).
- **Total Failed Collect**: the total amount of raw printing materials that were left or destroyed due to the collecting robot being infected.

- **Total Failed Print**: the total amount of printable components that were not fabricated due to the printing robot being infected.

The control experimental condition (A0) results are presented in Table 6.1. The results for experimental conditions T1 and T2 are presented in Tables 6.2-6.4. One table is included for each of the *step_infected* inputs (10, 20, and 40). The results for experimental condition T3 are presented in Table 6.5.

Table 6.1. Control condition results with base decision-making at time-step 70.

(Control Group)	**CHE**	**CHO**	**DHE**	**DHO**	**HHE**	**HHO**
Num Robots	29.2	28.2	137.4	83.8	165.5	142.4
Total Collected	500	500	500	499.3	500	500
Total Printed	68.9	0.0	550	548.1	550	549.9

Table 6.2. Results for condition where a robot gets infected at time-step 10.

Step_infected=10	***ID***	**CHE**	**CHO**	**DHE**	**DHO**	**HHE**	**HHO**
Num Robots	**T1**	28.8	28.4	111.6	31.8	132.2	82.5
	T2	29.3	28.0	88.3	31.4	102.2	56.7
Num Infected	**T1**	4.58	6.54	14.64	15.38	17.53	18.90
	T2	26.15	25.89	25.22	15.40	30.63	27.66
Num Infected Std Dev	**T1**	9.13	10.88	14.55	2.84	18.25	13.62
	T2	2.22	3.30	11.42	2.61	13.11	4.95
Total Failed Collect	**T1**	79.1	110.5	87.7	236.4	89.8	191.7
	T2	364.9	358.7	156.6	239.8	155.3	259.9
Total Failed Print	**T1**	13.8	0.0	71.4	151.7	73.3	87.5
	T2	0.0	0.0	99.5	151.3	117.4	132.6
Total Collected	**T1**	419.2	388.1	412.3	255.5	410.1	303.3
	T2	135.1	136.4	339.0	255.1	344.7	240.0
Total Printed	**T1**	54.7	0.5	390.9	151.7	386.9	265.8
	T2	68.4	0.6	289.1	151.4	277.3	157.1

Table 6.3. Results for condition where a robot gets infected at time-step 20.

Step_infected=20	***ID***	**CHE**	**CHO**	**DHE**	**DHO**	**HHE**	**HHO**
Num Robots	**T1**	29.6	28.1	130.7	67.8	159.1	123.7
	T2	29.5	28.2	125.7	67.5	149.3	109.5
Num Infected	**T1**	2.71	3.84	10.79	8.56	8.15	13.47
	T2	23.11	22.73	17.37	8.11	20.40	21.25
Num Infected Std Dev	**T1**	6.36	7.68	9.25	2.35	11.41	10.92
	T2	1.80	1.75	7.94	2.48	10.57	4.61
Total Failed Collect	**T1**	41.1	57.9	42.6	63.6	24.1	52.2
	T2	220.4	214.5	70.2	60.6	60.3	73.9
Total Failed Print	**T1**	3.0	0.0	35.0	61.5	32.9	61.5
	T2	0.0	0.1	34.7	58.3	47.2	100.4
Total Collected	**T1**	458.9	442.1	457.3	435.0	475.9	447.8
	T2	279.6	285.4	429.7	437.6	439.7	426.0
Total Printed	**T1**	65.9	0.2	472.3	422.2	493.0	436.3
	T2	68.9	0.0	445.1	427.4	442.5	375.6

Table 6.4. Results for condition where a robot gets infected at time-step 40.

Step_infected=40	***ID***	**CHE**	**CHO**	**DHE**	**DHO**	**HHE**	**HHO**
Num Robots	**T1**	28.9	28.0	138.8	82.2	164.3	143.4
	T2	28.8	28.7	138.4	80.4	166.0	142.4
Num Infected	**T1**	2.03	2.19	1.89	2.62	1.71	3.39
	T2	15.26	15.56	2.64	2.55	3.10	5.66
Num Infected Std Dev	**T1**	4.08	4.31	1.23	1.88	1.20	3.13
	T2	1.65	0.67	1.25	1.91	1.03	2.92
Total Failed Collect	**T1**	12.4	13.6	3.5	13.0	1.8	7.3
	T2	36.0	29.9	5.3	14.2	1.2	4.8
Total Failed Print	**T1**	2.1	0.3	3.5	16.0	2.6	11.2
	T2	0.0	0.0	0.3	14.0	0.9	19.9
Total Collected	**T1**	487.5	482.7	496.5	486.8	498.2	492.6
	T2	463.9	470.0	494.7	483.5	498.7	495.2
Total Printed	**T1**	66.5	0.3	543.0	520.4	545.6	531.4
	T2	68.9	0.0	544.4	517.9	547.9	525.2

Table 6.5. Results for experimental condition T3.

T3	**CHE**	**CHO**	**DHE**	**DHO**	**HHE**	**HHO**
Step_infected	*10*	*10*	*20.55*	*29.08*	*11*	*10*
Num Robots	29.1	28.5	139.4	79.9	165.7	142.9
Num Infected	0.59	0.70	0.92	2.55	0.59	0.55
Num Infected Std Dev	0.49	0.46	0.27	1.07	0.49	0.50
Total Failed Collect	31.4	35.6	6.7	12.3	24.4	30.2
Total Failed Print	0.0	0.0	19.9	18.9	0.0	0.0
Total Collected	465.6	464.3	493.3	486.8	475.6	469.8
Total Printed	68.9	0.0	523.4	516.6	525.6	519.8

6.4. Analysis

In this section, the results are analyzed and compared with the predictions hypothesized. Then, more general conclusions from the data are discussed. Figures 6.2-6.4 present graphs of the percentage of robots infected at the end of the simulation for each system configuration. Figures 6.5-6.7 present graphs of the percent reduction in the number of robots compared to the control experimental condition.

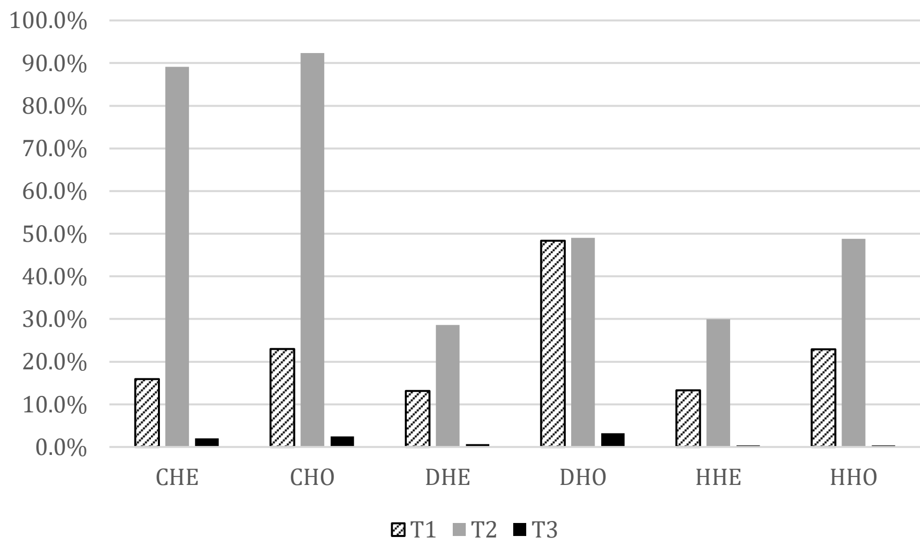


Figure 6.2. Percentage of robots infected when *step_infected*=10.

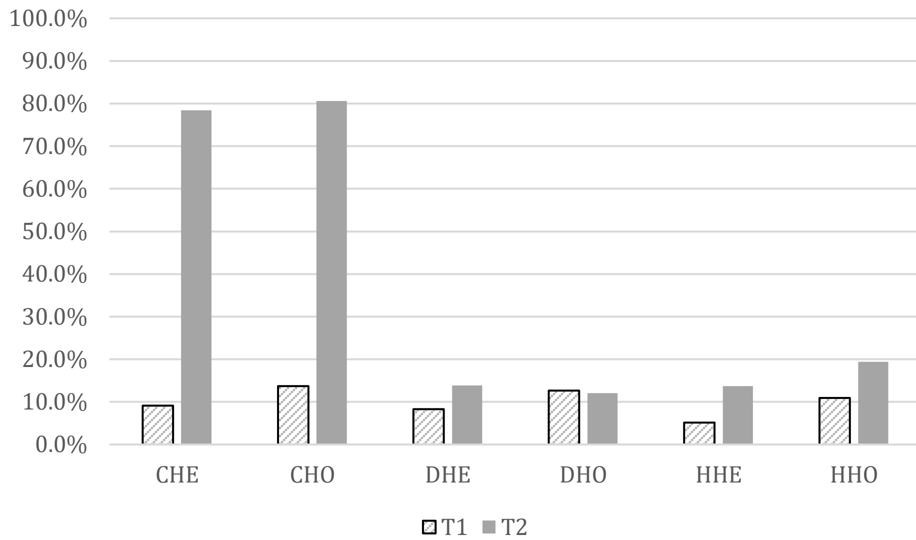


Figure 6.3. Percentage of robots infected when *step_infected*=20.

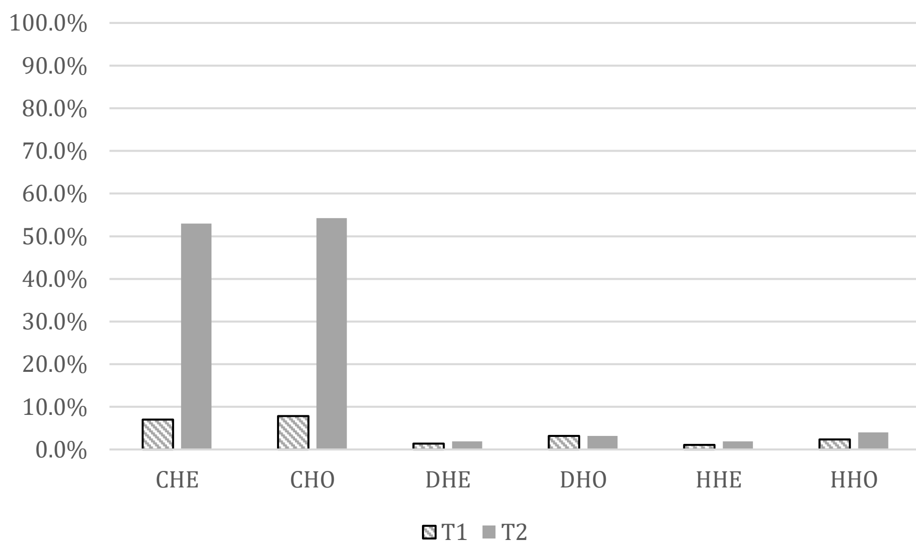
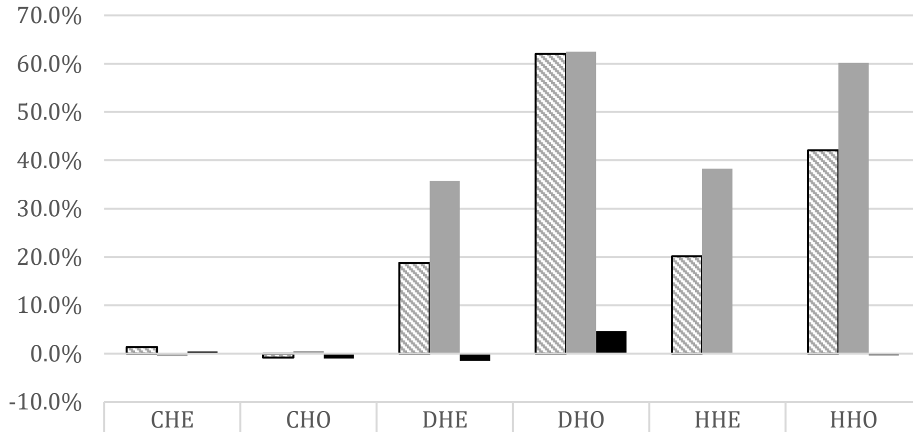
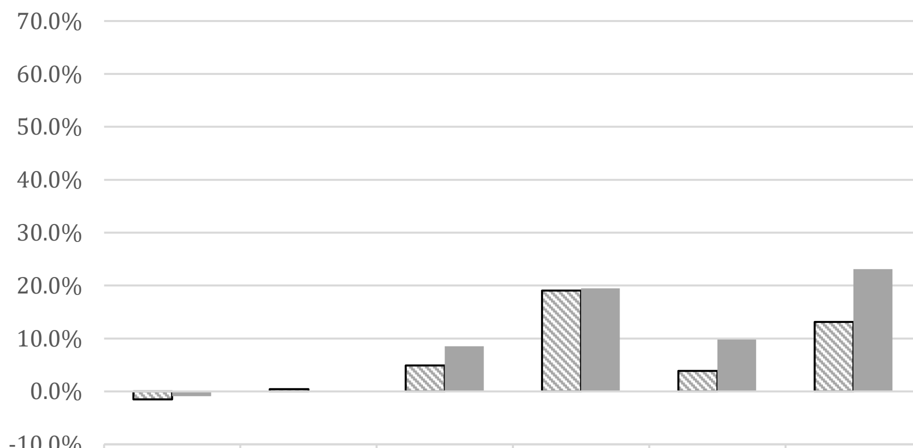


Figure 6.4. Percentage of robots infected when *step_infected*=40.



	CHE	CHO	DHE	DHO	HHE	HHO
T1	1.4%	-0.8%	18.8%	62.0%	20.1%	42.1%
T2	-0.5%	0.6%	35.7%	62.5%	38.3%	60.2%
T3	0.4%	-1.0%	-1.5%	4.7%	-0.1%	-0.4%

Figure 6.5. Percent decrease in number of robots when *step_infected*=10.



	CHE	CHO	DHE	DHO	HHE	HHO
T1	-1.5%	0.5%	4.9%	19.1%	3.9%	13.1%
T2	-0.9%	0.0%	8.5%	19.5%	9.8%	23.1%

Figure 6.6. Percent decrease in number of robots when *step_infected*=20.

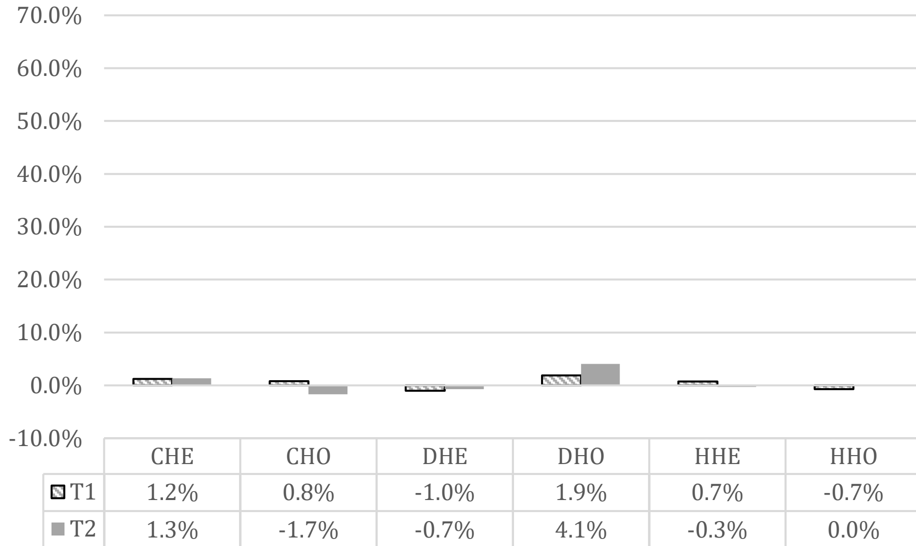


Figure 6.7. Percent decrease in number of robots when *step_infected*=40.

6.4.1. Evaluation of the Centralized Approach Hypothesis

The CHE and CHO configurations performed as hypothesized, in terms of the percentage of robots infected, under experimental conditions T1, T2, and T3. This is shown in Figures 6.2-6.4, where the CHE and CHO configurations had a significantly higher percentage of infected robots for experimental condition T2, but a lower percentage infected for experimental conditions T1 and T3. Furthermore, as shown in Tables 6.2-6.5, the amount of raw printing materials lost by the CHE and CHO configurations, due to infected robots, is pronounced for experimental condition T2, but negligible for experimental conditions T1 and T3. This supports the hypothesis that the targeted experimental condition (T2) impacts the centralized configurations most significantly.

The prediction that the CHE configuration would lose less fabricated components than the CHO configuration was not accurate. This outcome occurred because the CHO configuration had sufficient initial printable components to continue assembling, even at time-step 70. Furthermore, the CHE and CHO configurations did not experience a significant reduction in number of robots, as compared to the control experimental condition. This is potentially due to the same circumstance of having sufficient initial resources such that production was not impeded by significant resource loss.

6.4.2. Evaluation of the Decentralized Approach Hypothesis

The prediction that the decentralized configurations of DHE and DHO would be more greatly impacted by all of the experimental conditions than the hierarchical configurations of HHE and HHO, is unsupported by the results. The HHE and HHO configurations performed equivalent or better than the DHE configuration in terms of both the percentage of robots infected and the percentage decrease in the number of robots, as compared to the control condition (as shown in Figures 6.2-6.7). The prediction that the DHE configuration would be less impacted by experimental conditions T1 and T3, as compared to the DHO configuration, is supported by the results.

Contrary to prediction, the DHO configuration did not perform consistently under experimental condition T3 (but it did perform consistently for conditions T1 and T2). Based on the data presented in Table 6.5, this is due to the robots in the DHO configuration not collecting resources until time-step 29. In situations where the initial resources were decreased, this appears to have caused it to perform more consistently during experimental conditions T1 and T2.

6.4.3. Evaluation of the Hierarchical Approach Hypothesis

The prediction that the hierarchical configurations of HHE and HHO would be more affected by experimental condition T2, as compared to experimental conditions T1 and T3, is supported by the results. The HHE and HHO configurations had a small percentage of robots infected under experimental condition T3 (as shown in Figures 6.2-6.4). This supports the hypothesis. Furthermore, the prediction that the HHE configuration would be less affected under experimental conditions T1 and T3 than the HHO configuration, has limited support. This is due to experimental condition T3 having an insignificant impact on both configurations, which made the difference between the two negligible. However, the HHO configuration had a higher percentage of robots infected for experimental condition T1 than the HHE configuration did (as shown in Figures 6.2-6.4), and a higher percent reduction in robot numbers as compared to the control group (as shown in Figures 6.5-6.7).

Finally, the prediction that the HHE configuration would have an increased number of infected robots as compared to the other system configurations is not strongly supported by the results (presented in Tables 6.2-6.5). While it marginally outperformed the other tested configurations when the *step_infected* parameter was set to 10, this superior performance was not observed when this parameter was set to 20 or 40.

6.4.4. Discussion

Overall, the heterogeneous configurations of DHE and HHE had a lower percentage of infected robots and a lower percent reduction in number of robots, as compared to the homogeneous configurations of DHO and HHO. The DHE and HHE configurations also had a lower percentage of robots infected, as compared to the centralized configurations of CHE and CHO. However, the heterogeneous configurations had a higher percent decrease in number of robots than the centralized configurations of CHE and CHO for experimental conditions T1 and T2. This is because the CHE and CHO configurations produce less robots than the DHE and HHE configurations (and had sufficient resources to not be impeded by the resource loss that occurred). In this regard, the DHE and HHE configurations were more robust overall. Thus, under the experimental rules regarding how a rogue or infected robot may behave, that were used the hierarchical configurations of DHE and HHE were shown to be strong options for mitigating the potential negative effects.

6.5. Anomaly Detection System

In this section, the potential efficacy of an anomaly detection system is discussed. The modified form of the experiment (i.e., with the anomaly detection system) would involve monitoring usage and measurements of certain resources and attributes of the robot system and flagging something as suspicious if it deviates from the expected.

6.5.1. Deviation in Resource Acquisition

Monitoring for deviations in resource acquisition could be used across the entire robot system. Significant deviations from the expected values (i.e., the control group) may indicate that robots have been infected. This attribute won't necessarily be able to identify which robot is infected, but it could alert the system to a possible problem. Individual robot resource acquisition tasks could also be monitored. For many applications, this may be preferable as which robot is infected could be more discernable.

Under the current simulation design, infected robots that are tasked with collecting raw materials or fabricating materials into printable components fail to acquire the resource in every case. If individual robots can be monitored, this particular behavior of the infected robots could be very simply detected (this could also be used to detect if a robot malfunctions).

In the case that only the overall resource collection of the system is known at each time-step, then predicting the resource acquisition of the system and comparing it to the observed resource acquisition of the system may allow an anomaly detection system to detect deviations.

$$Predicted_{Collect}(t) = AssignedRobots_{Collect}(t) * Collect\_Amount$$
$$Predicted_{Print}(t) = AssignedRobots_{Print}(t) * Print\_Amount * Print\_Efficiency \qquad \text{(Eq. 5)}$$

In terms of the utilized simulation system, the amount of collected raw printing materials can be predicted at each time-step using Eq 5. In the equation, the $AssignedRobots_{TaskType}(t)$ variable refers to the number of robots assigned to perform a certain task type at time-step *t*. The error is then the observed amount of a resource acquired in the time-step minus the predicted amount. Figure 6.8 depicts the predicted and observed resources collected per time-step for the HHE configuration under experimental condition T2 and *step_infected* set to 10. This discrepancy is notable and could serve as a notification of a problem.

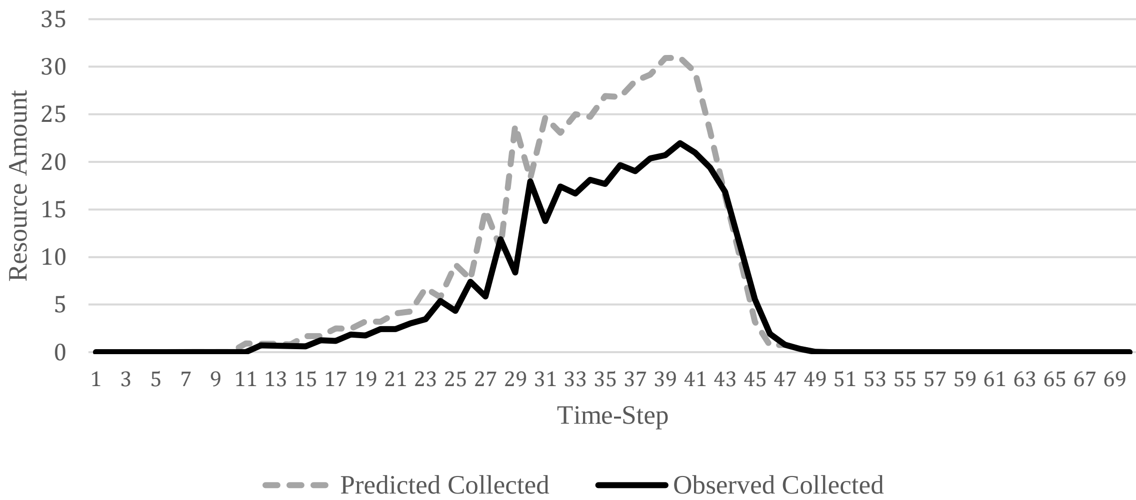


Figure 6.8. Resource acquisition for the HHE configuration on experimental condition T2.

6.5.2. Other Deviations

Some other characteristics that may be relevant, but for which data may not be collectible for in a simulation environment, include:

- **Changes in code**: This attribute may be the most obvious sign of tampering; however, it may not always be measurable. In addition, certain malicious additions may run in the background and not directly affect the main code of the robot. For the simulation, this includes not bringing resources to the stockpile and instead hiding or destroying them. This category also includes robots not performing the tasks assigned to them.
- **Significant design alterations**: This attribute is particular to the self-replicating capability. Here, if a robot that has been infected or tampered with builds a new robot,

the question of whether that built robot is also infected is a concern. This notion is used in the current simulation, and future work may involve giving infected robots deviations or other capabilities. Thus, if a robot were to exhibit capabilities that it shouldn't have, (i.e., those that a robot type would not have), then it would be suspicious.

- **Memory usage**: A significant enough increase in memory usage for a robot may indicate that malicious code is running, and the robot is thus using more memory than what would typically be expected.
- **Power consumption (for computing devices)**: An increase in power consumption for computing devices may be indicative of changes to code or additional processes running.
- **Processor usage**: Similar to memory and power usage, a significant change in processor usage may indicate that an unintended change in behavior has occurred.
- **Tampering with safeguard measures**: Safeguard measures that are in place to prevent certain unsafe or malicious activities would themselves need to be verified. Detection of a robot with any safeguards removed would be a sign of potential tampering.

6.6. Summary

In this chapter, cybersecurity considerations for self-replicating robot systems were discussed. An experiment was conducted to formulate a potential anomaly detection system for the application of self-replicating robots. The results of the experiment were presented and discussed.

System deviation from the expected behavior based on the simulation results from this chapter may provide a means of detecting if the system has been compromised. However, this would necessitate that the dynamic of what malicious acts or capabilities are carried out by the rogue robot(s) align with those outlined in the chapter. Further work and data are needed to further develop the proposed anomaly detection system.

7. SUMMARY AND CONCLUSION

In this chapter, a summary of the results presented in previous chapters is provided. Then, the overarching conclusions of this work are presented.

7.1. Simulation

In Chapter 3, the simulation system used for the experiments in subsequent chapters was detailed. There are four task types and four robot types in this simulation. The collect task type involves a robot gathering raw printing materials from the environment and adding the gathered materials to the robot system's inventory. The print task type involves a robot taking raw printing materials and crafting them into printable components. The assemble task type involves a robot taking nonprintable components and printable components from the robot system's resource pool and assembling them into a new robot. The idle task type is assigned to any robot not performing any other action during a time-step.

In the simulation, there are four types of robots: normal, printer, assembler, and replicator. In each time-step, each robot is either idle, gathering resources, printing components, or assembling a new robot. However, certain robot types are restricted in what types of tasks that they can perform. Robot capabilities are listed in Table 7.1. All robot types are capable of being idle. Not all types of robots must be included in any given simulation run.

Table 7.1. Robot types in the simulation.

Robot Type	*Collect Resources*	*Print Components*	*Assemble Robots*
Normal	●		
Assembler	●		●
Printer	●	●	
Replicator	●	●	●

There are six replication system configurations used in the simulation system. These system configurations vary in what types of robots they are able to build. These are derived from two sets of higher-level classifications being combined. The first, the replication approach, has three classifications. With the centralized approach, robots that have a replication-related capability are not buildable by the robot system. In the decentralized approach, all robots have one or more replication-related capabilities. Using the hierarchical approach, some robot types have replication-related capabilities, and some robot types do not.

A selection from this first set is then combined with a selection from a second set of production approaches. The results of combining the higher-level approaches are listed in Table 7.2. The second set, the production approach, has two classification values. In the homogeneous production approach, the robot system has a single robot type for all the replication-related capabilities. With the heterogeneous production approach, the robot system has multiple robot types that have replication-related capabilities.

Table 7.2. Robot system configurations.

System Configuration		**Robot Type**			
ID	**Name**	Normal	Printers	Assemblers	Replicators
CHO	Centralized Homogeneous	●	-	-	○
DHO	Decentralized Homogeneous	-	-	-	●
HHO	Hierarchical Homogeneous	●	-	-	●
CHE	Centralized Heterogeneous	●	○	○	-
DHE	Decentralized Heterogeneous	-	●	●	-
HHE	Hierarchical Heterogeneous	●	●	●	-

*Buildable robot types are denoted with the filled in circle, and robots that are present (but not buildable) are denoted with a hollow circle.

There are parameters in the simulation that set certain resource values, cost values, and environmental values which all influence the simulation. These parameters are varied in a series of experimental conditions.

7.2. System Configuration Experiment

An experiment was conducted using the simulation system to study the differences in performance caused by the system configurations of CHE, CHO, DHE, DHO, HHE, and HHO.

The centralized configurations, CHE and CHO, were shown to marginally outperform in high-risk cases, in terms of collection potential, as compared to other configurations. The heterogeneous configurations of DHE and HHE (not configuration CHE) were shown to reach a maximum number of robots (based on the available resources) more quickly than their homogeneous configuration counterparts. However, the homogeneous configurations, DHO and HHO, outperformed them in terms of assembly potential and print potential in later time-steps (for many experimental conditions). Finally, the decentralized configurations (DHE and DHO) outperformed their hierarchical counterparts in terms of assembly potential and print potential for most experimental conditions; however, they had a lower collection potential. Whether a system was a homogeneous or heterogeneous configuration was shown to be a more significant factor overall. These findings were used to establish the decision-making criteria for the decision-making algorithms.

7.3. Decision-Making Experiment

Three decision-making algorithms were also proposed and tested. These algorithms include the cycle, variable, and strategic decision-making algorithms. These were compared with the base algorithm and assessed based on their comparative performance. This experiment utilized the same 70 experimental conditions as the previous experiment.

In comparison to the base algorithm, the cycle decision-making algorithm had improved print potential for the DHE and HHE configurations in the later time-steps. However, the cycle-decision-making algorithm also had decreased assembly potential in most experimental conditions and time-steps.

The results from the base and cycle decision-making algorithms were used to construct the variable algorithm. This algorithm took an input of assembly, print, or collection potential and attempted to maximize it by selecting the highest scoring system configuration and decision-making combination from the base and cycle algorithms. However, it did not perform well in terms of collection potential and had problems in experimental conditions where the maximum number of robots was reached early.

Based on the variable algorithm's lack of sufficient improvements over the base and cycle decision-making algorithms, the strategic decision-making algorithm was devised. The variable algorithm can be characterized as a greedy algorithm, were the best performing option was selected without regard to looking ahead at subsequent steps. In contrast, the strategic algorithm has a time-step goal, in addition to the optimize-for input metric. Thus, its approach is to build a core number of robots in an initial phase, and then focus entirely on maximizing the input metric in the subsequent phase. This algorithm performed far better than the other algorithms in terms of the input metric for almost all experimental conditions and time-steps. Thus, the results show it to be a strong option for maximizing a particular capability metric. It often maximizes the input metric at the cost of other metrics. Thus, it is not suitable for use cases that require a more even distribution of metrics.

7.4. Cybersecurity

The cybersecurity considerations for self-replicating robot systems were also discussed. An experiment was conducted in order to formulate an intrusion detection system for self-replicating robot systems. The experiment analyzed the impact that the system configuration of the self-replicating system has on its cybersecurity vulnerabilities. The experiment was conducted by modifying the simulation from the previous experiments. In this experiment, a certain robot was infected at a specified time-step and the results were analyzed.

Overall, the heterogeneous configurations of DHE and HHE had a lower percentage of infected robots and a lower percent reduction in the number of robots, as compared to the homogeneous configurations of DHO and HHO. The DHE and HHE configurations also had a lower percentage of robots infected, as compared to the centralized configurations of CHE and CHO. However, the heterogeneous configurations had a higher percent decrease in the number of robots than the centralized configurations of CHE and CHO for experimental conditions T1 and T2. This was because the CHE and CHO configurations produced less robots than the DHE and HHE configurations (and had sufficient resources to not be impeded by resource loss). In this regard, the DHE and HHE configurations were more robust overall. Thus, under these particular experimental rules, regarding how a rogue or infected robot may behave, the hierarchical configurations of DHE and HHE were shown to be strong options for mitigating the potential negative effects.

System deviation from the expected behavior, based on the simulation results from this experiment, may provide a means of detecting if the system has been compromised. A simplistic anomaly detection system was proposed to detect deviations of resource acquisition values. However, this anomaly detection system only considered a subset of possible malicious acts. Further work and data are needed to further develop a robust intrusion detection system suitable for real-world use.

7.5. Conclusions

Self-replicating robots can be beneficial for use in areas that are difficult for humans to access or prohibitive to bring materials and supplies to. Beyond these specific areas of need, this type of robot system could theoretically be used for a wide variety of applications. An application domain that may especially benefit from self-replicating robots is aerospace. Launching materials into space can be prohibitively expensive, which may warrant utilizing in-situ materials.

The proposed simulation approach could be used with parameters that fit with known mission constraints in order to guide the decision of how the self-replicating robot system should be configured. Furthermore, the data from the simulation experiments may be useful for determining which types of buildable robots may be needed for a particular mission or objective. The data from the various experimental conditions in the experiments could potentially be used as a heuristic to optimize this choice based on the ratio of printable component costs to in-situ resource collection rate and the rate at which the in-situ resources are converted into printable components.

The initial resources provided to the system also impact the rate at which it expands. This could be adjusted based on the data in order to potentially fulfill objective related requirements. Furthermore, the risks associated with the various tasks can affect the viability and effectiveness of different approaches. If these are known or estimable, the simulation could be run with those parameters in order to plan a potentially effective system configuration to mitigate these risks.

The type of decision-making algorithm used for the system may also depend on system mission objectives. For instance, if maximizing the number of robots with a particular capability is required, then the strategic decision-making algorithm (or similar) may be useful for this task. This might be the case if the end goal was to print a larger structure and a number of 3D print-capable robots are needed. Similarly, if a larger structure required many assembly-capable robots then the end goal would align with that type of optimization. In contrast, if having a number of robots that are more diverse in their capabilities is preferred, then the more simplistic decision-making algorithms that use repeated build orders may be sufficient.

Finally, the threat of a rogue or infected robot in the robot system can be a concern. The effects of this may largely depend on the capabilities of the rogue or infected robot(s). The data presented in Chapter 6 may provide insight into which system configuration would potentially mitigate risks. If the expected malicious actions differ from those presented, then the simulation model itself could be used and adapted accordingly in order to better model the circumstances and identify deviations.

This dissertation has presented work on the command decision making for self-replicating robot systems, building on prior work (discussed in Section 2) regarding their mechanical and electronic designs. The use of self-replicating robot systems may benefit missions where safety or other considerations prohibit or reduce the desirability of using humans to complete mission goals. Particularly at long distances and in hard to access environments, the robots must have decision making capabilities. Autonomy is key to maximizing the efficiency of self-replicating robot systems and, in some cases, is key to being able to successfully complete a mission at all. The work presented herein is designed to benefit the implementation of such a robot system by modeling the multi-agent dynamic, heuristics for decision-making and providing an evaluation of effective command strategies for known or expected conditions.

REFERENCES

[1] M. Sipper, "Fifty Years of Research on Self-Replication: An Overview," *Artif. Life*, vol. 4, no. 3, pp. 237–257, 1998.

[2] A. Jones and J. Straub, "Concepts for 3D Printing-Based Self-Replicating Robot Command and Coordination Techniques," *Machines*, vol. 5, no. 2, Apr. 2017.

[3] R. A. Freitas and R. C. Merkle, *Kinematic self-replicating machines*. Georgetown, TX, USA: Landes Bioscience, 2004.

[4] Z. Yan, N. Jouandeau, and A. A. Cherif, "A Survey and Analysis of Multi-Robot Coordination," *Int. J. Adv. Robot. Syst.*, vol. 10, no. 12, Dec. 2013.

[5] A. Jones and J. Straub, "Self-replicating 3d printed satellites," in *Proceedings of the International Astronautical Congress, IAC*, 2017, vol. 13, pp. 8264–8271.

[6] B. J. Crespi and D. Yanega, "The definition of eusociality," *Behav. Ecol.*, vol. 6, no. 1, pp. 109–115, 1995.

[7] P. W. Sherman, E. A. Lacey, H. K. Reeve, and L. Keller, "Forum: The eusociality continuum," *Behav. Ecol.*, vol. 6, no. 1, pp. 102–108, 1995.

[8] B. Evans, *Practical 3D printers: The science and art of 3D printing*. 2012.

[9] T. Pfeifer, C. Koch, and L. Van Hulle, "OPTIMIZATION OF THE FDM™ ADDITIVE MANUFACTURING PROCESS," in *Proceedings of the Annual Technical Conference (ANTEC) of the Society of Plastics Engineers*, 2016.

[10] "Snappy." [Online]. Available: http://reprap.org/wiki/Snappy. [Accessed: 18-Jul-2017].

[11] J. Calì *et al.*, "3D-printing of non-assembly, articulated models," *ACM Trans. Graph.*, vol. 31, no. 6, p. 1, 2012.

[12] A. B. Jones and J. Straub, "Mission-Responsive. On-Demand 3D Printed Blimps for Martian Missions," in *Proceedings of the IEEE Aerospace Conference*, 2019, pp. 1–6.

[13] B. Matthias, S. Kock, H. Jerregard, M. Kallman, and I. Lundberg, "Safety of collaborative industrial robots: Certification possibilities for a collaborative assembly robot concept," in *2011 IEEE International Symposium on Assembly and Manufacturing (ISAM)*, 2011, pp. 1–6.

[14] W. Leyh, "Experiences with the construction of a building assembly robot," *Autom. Constr.*, vol. 4, no. 1, pp. 45–60, Mar. 1995.

[15] A. E. Quaid and R. L. Hollis, "Cooperative 2-DOF robots for precision assembly," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2188–2193.

[16] A. Bolger, M. Faulkner, D. Stein, L. White, Seung-kook Yun, and D. Rus, "Experiments in decentralized robot construction with tool delivery and assembly robots," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 5085–5092.

[17] A. F. Winfield, "Foraging Robots," in *Encyclopedia of Complexity and Systems Science*, vol. 6, New York, NY: Springer New York, 2009, pp. 3682–3700.

[18] B. Kading and J. Straub, "Utilizing in-situ resources and 3D printing structures for a manned Mars mission," *Acta Astronaut.*, vol. 107, pp. 317–326, 2015.

[19] G. Cesaretti, E. Dini, X. De Kestelier, V. Colla, and L. Pambaguian, "Building components for an outpost on the Lunar soil by means of a novel 3D printing technology," *Acta Astronaut.*, vol. 93, pp. 430–450, Jan. 2014.

[20] G. S. Chirikjian, Yu Zhou, and J. Suthakorn, "Self-replicating robots for lunar development," *IEEE/ASME Trans. Mechatronics*, vol. 7, no. 4, pp. 462–472, Dec. 2002.

[21] T. Campbell, C. Williams, B. Garrett, and O. Ivanova, “Could 3D Printing Change the World?,” *atltantic Counc.*, 2011.

[22] D. Periard, E. Malone, and H. Lipson, “Printing embedded circuits,” in *18th Solid Freeform Fabrication Symposium, SFF 2007*, 2007, pp. 503–512.

[23] J. Rossiter, P. Walters, and B. Stoimenov, “Printing 3D dielectric elastomer actuators for soft robotics,” in *Proceedings of SPIE*, 2009, vol. 7287, p. 72870H.

[24] A. Hatna, R. J. Grieve, and P. Broomhead, “Automatic CNC milling of pockets: geometric and technological issues,” *Comput. Integr. Manuf. Syst.*, vol. 11, no. 4, pp. 309–330, 1998.

[25] K. H. Waters, *Reflection seismology: A tool for energy resource exploration. Third edition*. 1987.

[26] T. N. A. of Sciences, *Evolutionary and Revolutionary Technologies for Mining.*, vol. 53, no. 9. Washington, D.C.: National Academies Press, 2002.

[27] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[28] J. J. Green and D. Vogt, “A Robot Miner for Low Grade Narrow Tabular Ore Bodies: The Potential and the Challenge,” *3rd Robot. Mechatronics Symp.*, 2009.

[29] P. A. Dunker, W. A. Lewinger, A. J. Hunt, and R. D. Quinn, “A Biologically Inspired Robot for Lunar Exploration and Regolith Excavation,” in *IEEE International Conference on Intelligent Robots and Systems*, 2009, pp. 5039–5044.

[30] M. Russell and J. Straub, “Software Design for an Intelligent Attitude Determination and Control System,” in *Research Experience for Undergraduates Poster Session at the AIAA/USU Conference on Small Satellites*, 2015.

[31] Timmiclark, "MR - 4 : Robotic Tank," *Thingiverse*, 2016. [Online]. Available: https://www.thingiverse.com/thing:1906831. [Accessed: 04-Mar-2020].

[32] J. J. Dunn *et al.*, "3D Printing in Space : Enabling New Markets and Accelerating the Growth of Orbital Infrastructure," in *Space Manufacturing 14: Critical Technologies for Space Settlement*, 2010, pp. 29–31.

[33] K. Zacny, P. Chu, J. Craft, and M. Cohen, "Asteroid mining," in *Proceedings of the AIAA SPACE 2013 conference and exposition*, 2013.

[34] C. Moore, "Technology Development for NASA's Asteroid Redirect Mission," in *65th International Astronautical Congress, IAC-14-D2*, 2014.

[35] J.-L. Beuchat and J. O. Haenni, "Von Neumann's 29-state cellular automaton: a hardware implementation," *IEEE Trans. Educ.*, vol. 43, no. 3, pp. 300–308, 2000.

[36] J. G. Kemeny, "Man Viewed as a Machine," *Sci. Am.*, vol. 192, no. 4, pp. 58–67, Apr. 1955.

[37] J. von Neumann, *The Theory of self reproducing automata*, 1st ed. Champaign, IL: University of Illinois Press, 1966.

[38] K. Lee, M. Moses, and G. S. Chirikjian, "Robotic Self-replication in Structured Environments: Physical Demonstrations and Complexity Measures," *Int. J. Rob. Res.*, vol. 27, no. 3–4, pp. 387–401, Mar. 2008.

[39] A. Sanderson, "Parts entropy methods for robotic assembly system design," in *Proceedings. 1984 IEEE International Conference on Robotics and Automation*, vol. 1, pp. 600–608.

[40] J. Suthakorn, Y. T. Kwon, and G. S. Chirikjian, "A semi-autonomous replicating robotic system," in *Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation.*, 2003, vol. 2, pp. 776–781.

[41] J. Suthakorn, Y. Zhou, and G. Chirikjian, “Self-Replicating Robots for Space Utilization,” in *Proceedings of the 2002 Robosphere Workshop on Self Sustaining Robotic Ecologies*, 2002, pp. 2–6.

[42] J. Suthakorn, A. B. Cushing, and G. S. Chirikjian, “An autonomous self-replicating robotic system,” in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, 2003, vol. 1, pp. 137–142.

[43] V. Zykov, E. Mytilinaios, B. Adams, and H. Lipson, “Robotics: self-reproducing machines.,” *Nature*, vol. 435, no. 7039, pp. 163–164, 2005.

[44] K. Gilpin and D. Rus, “What’s in the bag: A distributed approach to 3D shape duplication with modular robots,” *Robot. Sci. Syst.*, 2012.

[45] R. Jones *et al.*, “RepRap – the replicating rapid prototyper,” *Robotica*, vol. 29, no. January 2011, pp. 177–191, 2011.

[46] G. M. Whitesides, “Self-Assembly at All Scales,” *Science (80-.).*, vol. 295, no. 5564, pp. 2418–2421, Mar. 2002.

[47] E. Tuci and M. Dorigo, “Cooperation Through Self-Assembly in Multi-Robot Systems,” *ACM Trans. Auton. Adapt. Syst.*, vol. 1, no. 2, pp. 115–150, 2006.

[48] V. Zykov, E. Mytilinaios, M. Desnoyer, and H. Lipson, “Evolved and Designed Self-Reproducing Modular Robotics,” *IEEE Trans. Robot.*, vol. 23, no. 2, pp. 308–319, Apr. 2007.

[49] Z. Butler, S. Murata, and D. Rus, “Distributed Replication Algorithms for Self-Reconfiguring Modular Robots,” in *Distributed Autonomous Robotic Systems 5 (DARS’02)*, 2002, pp. 37–48.

[50] E. Sahin *et al.*, “SWARM-BOT: pattern formation in a swarm of self-assembling mobile robots,” in *IEEE International Conference on Systems, Man and Cybernetics*, 2002, vol. vol.4, p. 6.

[51] R. Groß, M. Dorigo, and M. Yamakita, “Self-assembly of Mobile Robots: From Swarm-bot to Super-mechano Colony,” in *Intelligent Autonomous Systems 9*, 2003, pp. 487–496.

[52] A. Jones and J. Straub, “Student Benefits from Participation in a NASA-mentored 3D Printing Research Project,” in *Proceedings of the Midwest Instruction and Computing Symposium*, 2018.

[53] C. K. Chua and K. F. Leong, *3D printing and additive manufacturing : principles and applications*. 2015.

[54] B. Berman, “3-D printing: The new industrial revolution,” *Bus. Horiz.*, vol. 55, no. 2, pp. 155–162, 2012.

[55] C. Schubert, M. C. van Langeveld, and L. A. Donoso, “Innovations in 3D printing: a 3D overview from optics to organs,” *Br. J. Ophthalmol.*, vol. 98, no. 2, pp. 159–161, 2014.

[56] A. B. Stroud, M. Morris, K. Carey, J. C. Williams, C. Randolph, and A. B. Williams, “MU-L8 : The Design Architecture and 3D Printing of a Teen-Sized Humanoid Soccer Robot,” 2013.

[57] P. Walters and D. McGoran, “Digital fabrication of ‘ smart ’ structures and mechanisms - creative applications in art and design,” *Int. Conf. Digit. Print. Technol. Digit. Fabr. 2011*, pp. 185–188, 2011.

[58] J. Hiller and H. Lipson, “Automatic Design and Manufacture of Soft Robots,” *IEEE Trans. Robot.*, vol. 28, no. 2, pp. 457–466, Apr. 2012.

[59] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, May 2015.

[60] N. W. Bartlett *et al.*, "A 3D-printed, functionally graded soft robot powered by combustion," *Science (80-.).*, vol. 349, no. 6244, pp. 161–165, Jul. 2015.

[61] R. Pfeifer, M. Lungarella, and F. Iida, "Self-Organization, Embodiment, and Biologically Inspired Robotics," *Science (80-.).*, vol. 318, no. 5853, pp. 1088–1093, Nov. 2007.

[62] R. Fisher, "The genetical theory of natural selection," *Dover Publ.*, vol. 22, pp. 127–130, 1958.

[63] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intell.*, vol. 7, no. 1, pp. 1–41, Mar. 2013.

[64] M. H. Dickinson, "Bionics: biological insight into mechanical design.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 96, no. 25, pp. 14208–14209, 1999.

[65] H. Klauk, *Organic Electronics: Materials, Manufacturing and Applications*. 2006.

[66] J. E. Ferrell, "Self-perpetuating states in signal transduction: positive feedback, double-negative feedback and bistability," *Curr. Opin. Cell Biol.*, vol. 14, no. 2, pp. 140–148, Apr. 2002.

[67] P. L. Aukrust, J. K. Dama˙, N. O. Solum, and † Oslo, "Soluble CD40 ligand and platelets: self-perpetuating pathogenic loop in thrombosis and inflammation?," *J. Am. Coll. Cardiol.*, vol. 43, no. 12, pp. 2326–2328, Jun. 2004.

[68] H. Ylihärsilä, J. G. Eriksson, T. Forsén, E. Kajantie, C. Osmond, and D. J. P. Barker, "Self-perpetuating effects of birth size on blood pressure levels in elderly people," *Hypertension*, vol. 41, no. 3 I, pp. 446–450, Mar. 2003.

[69] N. Zilka, B. Kovacech, P. Barath, E. Kontsekova, and M. Novák, “The self-perpetuating tau truncation circle,” in *Biochemical Society Transactions*, 2012, vol. 40, no. 4, pp. 681–686.

[70] C. Alexakis, T. Partridge, and G. Bou-Gharios, “Implication of the satellite cell in dystrophic muscle fibrosis: A self-perpetuating mechanism of collagen overproduction,” *Am. J. Physiol. - Cell Physiol.*, vol. 293, no. 2, pp. 661–669, 2007.

[71] J. Ma and S. Lindquist, “Conversion of PrP to a self-perpetuating PrPSc-like conformation in the cytosol,” *Science (80-.).*, vol. 298, no. 5599, pp. 1785–1788, 2002.

[72] J. C. W. Edwards, G. Cambridge, and V. M. Abrahams, “Do self-perpetuating B lymphocytes drive human autoimmune disease?,” *Immunology*, vol. 97, no. 2, pp. 188–196, Jun. 1999.

[73] N. ben Ming, K. Tsukamoto, I. Sunagawa, and A. A. Chernov, “Stacking faults as self-perpetuating step sources,” *J. Cryst. Growth*, vol. 91, no. 1–2, pp. 11–19, Aug. 1988.

[74] J. A. Nuth, N. M. Johnson, and S. Manning, “A self-perpetuating catalyst for the production of complex organic molecules in protostellar nebulae,” in *Proceedings of the International Astronomical Union*, 2008, vol. 4, no. S251, pp. 403–408.

[75] E. D’Onghia, M. Vogelsberger, and L. Hernquist, “Self-perpetuating spiral arms in disk galaxies,” *Astrophys. J.*, vol. 766, no. 1, p. 34, 2013.

[76] J. M. Beer, A. D. Fisk, and W. A. Rogers, “Toward a Framework for Levels of Robot Autonomy in Human-Robot Interaction,” *J. Human-Robot Interact.*, vol. 3, no. 2, p. 74, 2014.

[77] A. Lampe and R. Chatila, “Performance measure for the evaluation of mobile robot autonomy,” in *IEEE International Conference on Robotics and Automation*, 2006, pp. 4057–4062.

[78] J. Yang and J. F. Coughlin, “In-vehicle technology for self-driving cars: Advantages and challenges for aging drivers,” *Int. J. Automot. Technol.*, vol. 15, no. 2, pp. 333–340, Mar. 2014.

[79] A. Xu, A. Kalmbach, and G. Dudek, “Adaptive Parameter EXploration (APEX): Adaptation of robot autonomy from human participation,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2014, pp. 3315–3322.

[80] G. Shaffer and A. Stentz, “A robotic system for underground coal mining,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 1992, vol. 1, pp. 633–638.

[81] C. Baldassano, “Explore vs. exploit: Task allocation for multi-robot foraging,” *Preprint*, 2009.

[82] Y. Cai, “Intelligent Multi-robot Cooperation for Target Searching and Foraging Tasks in Completely Unknown Environments,” University of Guelph, 2013.

[83] M. S. Fibla, U. Bernardet, and P. F. M. J. Verschure, “Allostatic control for robot behaviour regulation: An extension to path planning,” in *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, 2010, no. October 2015, pp. 1935–1942.

[84] J. P. Hecker, J. C. Carmichael, and M. E. Moses, “Exploiting clusters for complete resource collection in biologically-inspired robot swarms,” in *IEEE International Conference on Intelligent Robots and Systems*, 2015, vol. 2015-Decem, pp. 434–440.

[85] J. Iqbal, R. U. Islam, S. Z. Abbas, A. A. Khan, and S. A. Ajwad, “Automating industrial tasks through mechatronic systems – a review of robotics in industrial perspective,” *Teh. Vjesn. - Tech. Gaz.*, vol. 23, no. 3, pp. 917–924, Jun. 2016.

[86] P. Leitão, A. W. Colombo, and S. Karnouskos, “Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges,” *Comput. Ind.*, vol. 81, pp. 11–25, Sep. 2016.

[87] M. A. Saliba, D. Zammit, and S. Azzopardi, “Towards practical, high-level guidelines to promote company strategy for the use of reconfigurable manufacturing automation,” *Robot. Comput. Integr. Manuf.*, vol. 47, pp. 53–60, 2017.

[88] B. Sadrfaridpour, H. Saeidi, J. Burke, K. Madathil, and Y. Wang, “Modeling and Control of Trust in Human-Robot Collaborative Manufacturing,” in *Robust Intelligence and Trust in Autonomous Systems*, Boston, MA: Springer US, 2016, pp. 115–141.

[89] B. Vogel-Heuser, D. Schütz, T. Frank, and C. Legat, “Model-driven engineering of Manufacturing Automation Software Projects - A SysML-based approach,” *Mechatronics*, vol. 24, no. 7, pp. 883–897, 2014.

[90] V. Vyatkin, “Software Engineering in Industrial Automation: State-of-the-Art Review,” *IEEE Trans. Ind. Informatics*, vol. 9, no. 3, pp. 1234–1249, Aug. 2013.

[91] C. Nieto-Granda, J. G. Rogers, and H. I. Christensen, “Coordination strategies for multi-robot exploration and mapping,” *Int. J. Rob. Res.*, vol. 33, no. 4, pp. 519–533, 2014.

[92] D. Portugal and R. P. Rocha, “Distributed multi-robot patrol: A scalable and fault-tolerant framework,” *Rob. Auton. Syst.*, vol. 61, no. 12, pp. 1572–1587, 2013.

[93] H. Liu, N. Stoll, S. Junginger, and K. Thurow, “Mobile robot for life science automation,” *Int. J. Adv. Robot. Syst.*, vol. 10, pp. 1–14, 2013.

[94] A. Pennisi *et al.*, “Multi-robot Surveillance Through a Distributed Sensor Network,” in *Studies in Computational Intelligence*, vol. 604, 2015, pp. 77–98.

[95] G. Starke, D. Hahn, D. Pedroza Yanez, and L. Ugalde Leal, "Self-Organization and Self-Coordination in Welding Automation with Collaborating Teams of Industrial Robots," *Machines*, vol. 4, no. 4, p. 23, Nov. 2016.

[96] I. Caliskanelli, B. Broecker, and K. Tuyls, "Multi-Robot Coverage: A Bee Pheromone Signalling Approach," in *Communications in Computer and Information Science*, vol. 519, 2015, pp. 124–140.

[97] J. Straub, "A Distributed Blackboard Approach Based Upon a Boundary Node Concept," *J. Intell. Robot. Syst.*, vol. 82, no. 3–4, pp. 467–478, Jun. 2016.

[98] B. J. Julian, M. Angermann, M. Schwager, and D. Rus, "Distributed robotic sensor networks: An information-theoretic approach," *Int. J. Rob. Res.*, vol. 31, no. 10, pp. 1134–1154, 2012.

[99] J. Zhou, X. Wu, and Z. Liu, "Distributed coordinated adaptive tracking in networked redundant robotic systems with a dynamic leader," *Sci. China Technol. Sci.*, vol. 57, no. 5, pp. 905–913, 2014.

[100] I. Navarro and F. Matía, "An Introduction to Swarm Robotics," *ISRN Robot.*, vol. 2013, pp. 1–10, 2013.

[101] E. Sahin, "Swarm Robotics: From Sources of Inspiration to Domains of Application," in *Swarm robotics*, Springer Berlin Heidelberg, 2005, pp. 10–20.

[102] T. AbuKhalil, T. Sobh, and M. Patil, "Survey on Decentralized Modular Robots and Control Platforms," *Int. J. Eng.*, vol. 7, no. 2, pp. 44–60, 2013.

[103] J. C. Barca and Y. A. Sekercioglu, "Swarm robotics reviewed," *Robotica*, vol. 31, no. 3, pp. 345–359, May 2013.

[104] Y. Altshuler, A. M. Bruchstein, and I. A. Wagner, "Swarm robotics for a dynamic cleaning problem," in *Proceedings - 2005 IEEE Swarm Intelligence Symposium, SIS 2005*, 2005, vol. 2005, pp. 217–224.

[105] Y. Mohan and S. G. Ponnambalam, "An extensive review of research in swarm robotics," in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, 2009, pp. 140–145.

[106] A. Campo and M. Dorigo, "Efficient Multi-foraging in Swarm Robotics," in *Advances in Artificial Life*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 696–705.

[107] F. Mondada, L. M. Gambardella, D. Floreano, S. Nolfi, J. L. Deneubourg, and M. Dorigo, "The cooperation of swarm-bots: Physical interactions in collective robotics," *IEEE Robot. Autom. Mag.*, vol. 12, no. 2, pp. 21–28, 2005.

[108] B. P. Gerkey, "ON MULTI-ROBOT TASK ALLOCATION," University of Southern California, 2003.

[109] J. D. Cannady, "Artificial neural networks for misuse detection," in *Proceedings of the 21st National information systems security conference*, 1998, pp. 368–381.

[110] L. Portnoy, E. Eskin, and S. Stolfo, "Intrusion detection with unlabeled data using clustering," in *Proceedings of ACM CSS Workshop on Data Mining Applied to Security*, 2001, pp. 1–25.

[111] J. Ryan, M. J. Lin, and R. Miikkulainen, "Iintrusion detection with neural networks," in *Advances in Neural Information Processing Systems*, 1998, pp. 943–949.

[112] A. Jones and J. Straub, "Using deep learning to detect network intrusions and malware in autonomous robots," in *SPIE Defense + Security*, 2017, vol. 10185.

[113] T. Bonaci and H. J. Chizeck, “On potential security threats against rescue robotic systems,” in *2012 IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR 2012*, 2012.

[114] A. Y. Javaid, W. Sun, V. K. Devabhaktuni, and M. Alam, “Cyber security threat analysis and modeling of an unmanned aerial vehicle system,” in *2012 IEEE Conference on Technologies for Homeland Security (HST)*, 2012, pp. 585–590.

[115] J. Schumann, P. Moosbrugger, and K. Y. Rozier, “R2U2: Monitoring and Diagnosis of Security Threats for Unmanned Aerial Systems,” *Runtime Verification*. p. 15, 2015.

[116] F. Higgins, A. Tomlinson, and K. M. Martin, “Threats to the Swarm : Security Considerations for Swarm Robotics,” *Int. J. Adv. Secur.*, vol. 2, no. 2&3, pp. 288–297, 2009.

APPENDIX A. SYSTEM CONFIGURATION RESULT TABLES

A.1. Base Algorithm: Time-Step 30

In this subsection, the assembly, print, and collection potentials are provided for time-step 30. Table A.1 lists the data for assembly potential. Table A.2 lists the data for print potential. Table A.3 lists the data for the collection potential.

Table A.1. Assembly potential and standard deviation for time-step 30.

Base Algorithm	ID	Assembly Potential						Assembly Potential Std Dev					
Time-Step: 30		CHE	CHO	DHE	DHO	HHE	HHO	CHE	CHO	DHE	DHO	HHE	HHO
(Default)	A0	0.99	1.00	22.67	15.70	18.35	13.56	0.10	0.00	1.46	0.69	2.15	1.03
BaseCost_Pr + 1	A1	1.00	0.99	18.57	15.52	17.99	11.65	0.00	0.10	1.33	0.94	1.21	0.90
BaseCost_Pr + 3	A2	0.99	0.99	16.58	13.47	15.39	8.92	0.10	0.10	1.07	1.18	1.56	0.39
BaseCost_Pr + 5	A3	1.00	0.99	13.23	11.70	10.59	9.80	0.00	0.10	1.24	0.67	1.30	0.72
PrintCost_Pr + 1	A4	1.00	1.00	20.23	15.38	18.33	12.77	0.00	0.00	1.29	1.42	1.08	0.71
PrintCost_Pr + 3	A5	1.00	1.00	19.67	13.57	18.02	10.86	0.00	0.00	1.41	0.86	1.25	0.53
PrintCost_Pr + 5	A6	0.99	1.00	21.24	11.75	16.87	8.89	0.10	0.00	1.68	0.56	1.49	0.42
AssembleCost_Pr + 1	A7	1.00	1.00	19.73	15.40	18.36	12.84	0.00	0.00	1.10	1.22	1.03	0.47
AssembleCost_Pr + 3	A8	0.98	0.99	16.71	13.59	16.96	10.76	0.14	0.10	1.30	0.85	1.25	0.71
AssembleCost_Pr + 5	A9	1.00	1.00	14.97	11.73	12.94	8.79	0.00	0.00	1.73	0.60	1.05	0.57
BaseCost_Time + 2	A10	1.00	1.00	7.81	7.85	6.89	6.78	0.00	0.00	0.54	0.54	0.35	0.66
BaseCost_Time - 1	A11	0.99	1.00	34.22	23.76	30.20	17.70	0.10	0.00	2.60	1.56	2.14	0.72
PrintCost_Time + 2	A12	1.00	1.00	20.05	7.88	16.59	7.82	0.00	0.00	1.26	0.48	0.83	0.67
PrintCost_Time - 1	A13	0.99	0.98	26.03	23.77	24.14	16.56	0.10	0.14	1.70	1.44	2.86	0.94
AssembleCost_Time + 2	A14	1.00	1.00	11.71	7.81	8.89	7.67	0.00	0.00	0.94	0.56	0.42	1.01
AssembleCost_Time - 1	A15	0.99	1.00	31.79	23.74	27.35	16.59	0.10	0.00	2.05	1.28	1.40	0.88
Print & Assemble Time + 2	A16	1.00	1.00	7.79	3.87	7.89	3.95	0.00	0.00	0.59	0.53	0.45	0.30
BaseCost_NonPr + 1	A17	1.00	1.00	22.64	15.34	18.70	13.51	0.00	0.00	1.40	1.47	0.73	1.18
PrintCost_NonPr + 1	A18	0.99	1.00	22.79	15.04	18.74	13.64	0.10	0.00	1.14	2.30	0.68	0.85
AssembleCost_NonPr + 1	A19	1.00	0.99	22.71	15.41	18.47	13.68	0.00	0.10	1.30	1.18	1.04	1.01
[All]CostPrintable + 1	A20	1.00	1.00	17.77	13.66	15.80	9.87	0.00	0.00	1.57	0.79	1.17	0.49
[All]CostPrintable + 2	A21	1.00	1.00	15.29	10.70	12.69	7.81	0.00	0.00	1.39	0.67	1.10	0.44
Print & Assemble Pr + 2	A22	1.00	1.00	18.12	12.48	16.70	9.80	0.00	0.00	1.35	0.90	1.32	0.53
Base & Print Pr + 2	A23	1.00	1.00	19.02	12.56	15.83	8.87	0.00	0.00	1.66	0.99	1.21	0.49
Base & Assemble Pr + 2	A24	1.00	0.99	16.07	12.56	12.85	8.96	0.00	0.10	1.37	1.05	1.21	0.20
[All]CostPrintable - 1	A25	1.00	1.00	26.00	15.40	18.49	13.45	0.00	0.00	1.95	1.15	1.08	1.27
AssembleCost_Pr - 1	A26	1.00	1.00	25.59	15.59	18.47	13.60	0.00	0.00	1.98	0.91	1.11	0.94
PrintCost_Pr - 1	A27	0.99	1.00	24.78	15.41	18.69	13.59	0.10	0.00	1.88	1.89	0.73	0.87
BaseCost_Pr - 1	A28	0.99	0.99	25.84	15.44	18.66	13.46	0.10	0.10	1.61	1.34	0.71	1.19
Print_Efficiency = 0.25	B1	0.99	1.00	16.42	15.29	14.51	13.71	0.10	0.00	1.14	1.81	1.15	0.87
Print_Efficiency = 0.5	B2	1.00	0.99	17.73	15.58	17.90	13.58	0.00	0.10	1.32	0.83	1.35	1.06
Print_Efficiency = 1.5	B3	1.00	1.00	25.79	15.55	18.48	13.69	0.00	0.00	1.64	1.28	1.15	0.96
Collect_Amount = 0.25	B4	1.00	0.98	21.37	15.64	17.88	13.54	0.00	0.14	1.46	0.86	1.30	0.91

(continues)

Table A.1. Assembly potential and standard deviation for time-step 30 (continued).

Base Algorithm	ID	Assembly Potential						Assembly Potential Std Dev					
Time-Step: 30		CHE	CHO	DHE	DHO	HHE	HHO	CHE	CHO	DHE	DHO	HHE	HHO
Collect_Amount = 0.5	B5	1.00	1.00	21.71	15.44	18.51	13.66	0.00	0.00	1.53	1.84	0.83	0.92
Collect_Amount = 1.5	B6	1.00	1.00	23.44	15.49	18.48	13.67	0.00	0.00	1.60	1.21	0.92	0.94
Print_Amount = 0.25	B7	1.00	1.00	18.28	15.60	14.80	13.54	0.00	0.00	1.44	0.95	1.06	1.15
Print_Amount = 0.5	B8	1.00	1.00	19.55	15.71	17.83	13.67	0.00	0.00	1.40	0.73	2.09	0.91
Print_Amount = 1.5	B9	0.98	0.99	24.10	15.39	18.53	13.42	0.14	0.10	1.36	1.28	1.11	1.44
Collect & Print Amount = 0.5	B10	1.00	0.99	19.62	15.44	17.98	13.42	0.00	0.10	1.24	1.26	1.52	1.84
QualityThreshold + 0.1	C1	1.00	1.00	22.11	15.05	17.78	13.16	0.00	0.00	1.71	1.62	1.62	1.39
QualityThreshold + 0.2	C2	1.00	1.00	21.45	13.97	17.21	12.25	0.00	0.00	2.27	1.91	2.56	1.88
QualityThreshold + 0.3	C3	1.00	1.00	17.58	11.57	13.46	10.57	0.00	0.00	4.23	2.89	3.14	2.10
QualityThreshold + 0.4	C4	1.00	1.00	12.01	8.12	9.83	7.03	0.00	0.00	4.32	2.93	3.00	2.68
RiskAmount_Print = 1%	C5	1.00	1.00	22.65	15.51	18.30	13.54	0.00	0.00	1.57	0.88	1.18	1.22
RiskAmount_Assemble = 1%	C6	0.98	0.96	20.17	12.25	14.35	11.42	0.14	0.20	2.60	3.49	2.90	2.45
RiskAmount Pr & As = 10%	C7	0.73	0.75	3.83	2.38	2.88	2.31	0.45	0.44	2.53	1.64	1.78	1.58
RiskAmount Pr & As = 15%	C8	0.67	0.59	2.01	1.28	1.80	1.32	0.47	0.49	1.76	1.31	1.53	1.27
RiskAmount_Assemble = 15%	C9	0.70	0.62	1.89	1.20	1.96	1.27	0.46	0.49	1.66	1.13	1.51	1.00
Quality_incr_Chance = 0.01%	C10	1.00	1.00	22.30	15.33	18.39	13.58	0.00	0.00	2.05	1.45	1.42	0.82
Quality_decr_Chance = 25%	C11	1.00	1.00	23.14	15.66	18.64	13.93	0.00	0.00	0.98	1.04	0.94	0.38
Quality_decr_Chance = 75%	C12	1.00	0.99	21.91	15.15	18.05	13.31	0.00	0.10	1.86	1.31	1.40	1.01
Quality_decr_Upper = 0.5	C13	1.00	1.00	19.81	12.75	16.60	12.14	0.00	0.00	2.55	2.69	2.12	1.65
Qual_incr Chance & Upper * 2	C14	1.00	1.00	22.35	15.60	18.46	13.67	0.00	0.00	2.67	0.83	1.12	0.68
RiskQuality_Modifier = 10.0	C15	1.00	1.00	22.45	15.43	18.02	13.58	0.00	0.00	1.70	1.21	1.79	1.03
RiskQuality_Modifier = 25.0	C16	1.00	1.00	21.30	14.86	17.90	13.49	0.00	0.00	1.76	2.18	1.57	1.30
RiskFactory_Modifier = 0.5	C17	0.99	0.99	22.81	15.61	18.22	13.61	0.10	0.10	1.20	0.82	1.74	0.93
RiskFactory_Modifier = 1.0	C18	0.96	0.95	22.33	15.35	18.10	13.51	0.20	0.22	2.06	1.27	2.20	1.26
Quality Thres & Chance	C19	1.00	1.00	7.41	5.17	6.11	5.12	0.00	0.00	3.26	2.46	2.34	2.10
Initial_Printable / 2.0	D1	1.00	1.00	18.67	12.70	17.21	7.93	0.00	0.00	1.19	0.72	1.08	0.43
Initial_Printable * 2.0	D2	1.00	0.99	25.91	15.38	18.41	13.48	0.00	0.10	1.90	1.36	1.22	1.07
Initial_Materials = 0	D3	1.00	0.99	19.90	14.72	18.41	13.69	0.00	0.10	1.79	1.87	1.36	0.83
Initial_Materials / 2.0	D4	1.00	1.00	20.82	15.20	18.38	13.55	0.00	0.00	1.22	1.28	1.15	1.00
Initial_Materials * 2.0	D5	1.00	1.00	25.80	15.52	18.53	13.70	0.00	0.00	2.84	1.19	0.88	0.72
Env_Materials / 2.0	D6	0.99	1.00	22.72	15.37	18.44	13.70	0.10	0.00	1.23	1.29	1.09	0.93
Env_Materials * 2.0	D7	1.00	1.00	22.70	15.69	18.49	13.54	0.00	0.00	1.32	0.79	1.13	1.21
Env_Materials * 100	D8	1.00	1.00	22.73	15.40	18.44	13.38	0.00	0.00	1.24	1.29	1.16	1.21
Initial_NonPr / 2.0	D9	1.00	1.00	22.56	15.57	18.59	13.76	0.00	0.00	1.57	0.82	1.03	0.67
Initial_NonPr * 2.0	D10	1.00	1.00	22.57	15.22	18.37	13.71	0.00	0.00	1.30	1.99	1.20	0.73
Initial NonPr & Env * 2.0	D11	1.00	1.00	22.43	15.37	18.31	13.53	0.00	0.00	1.74	1.47	1.40	1.18
Initial NonPr & Env * 2.0, Raw=0	D12	1.00	1.00	19.44	15.06	18.56	13.56	0.00	0.00	2.51	1.27	1.12	1.00

Table A.2. Print potential and standard deviation for time-step 30.

Base Algorithm	ID	Print Potential						Print Potential Std Dev					
Time-Step: 30		CHE	CHO	DHE	DHO	HHE	HHO	CHE	CHO	DHE	DHO	HHE	HHO
(Default)	A0	1.00	1.00	18.36	15.66	13.57	13.67	0.00	0.00	1.52	0.65	1.44	0.71
BaseCost_Pr + 1	A1	0.99	1.00	14.11	15.58	12.69	11.73	0.10	0.00	0.99	0.75	0.72	0.63
BaseCost_Pr + 3	A2	1.00	1.00	9.55	13.45	6.86	8.88	0.00	0.00	0.80	1.08	0.47	0.36
BaseCost_Pr + 5	A3	1.00	1.00	5.89	11.59	4.94	9.78	0.00	0.00	0.35	0.70	0.37	0.63
PrintCost_Pr + 1	A4	1.00	1.00	16.43	15.43	13.59	12.79	0.00	0.00	1.30	1.17	0.68	0.59
PrintCost_Pr + 3	A5	0.99	1.00	12.96	13.55	12.64	10.88	0.10	0.00	1.13	0.80	0.85	0.38
PrintCost_Pr + 5	A6	1.00	1.00	10.46	11.73	10.61	8.94	0.00	0.00	0.72	0.51	0.80	0.24
AssembleCost_Pr + 1	A7	1.00	1.00	16.45	15.37	13.60	12.86	0.00	0.00	1.20	1.08	0.82	0.35
AssembleCost_Pr + 3	A8	1.00	1.00	14.14	13.55	10.86	10.82	0.00	0.00	1.28	0.77	0.57	0.50
AssembleCost_Pr + 5	A9	1.00	1.00	11.33	11.63	11.80	8.91	0.00	0.00	1.25	0.63	0.45	0.35
BaseCost_Time + 2	A10	0.98	1.00	4.76	7.92	4.91	6.87	0.14	0.00	0.49	0.37	0.32	0.42
BaseCost_Time - 1	A11	1.00	1.00	27.98	23.82	23.87	17.68	0.00	0.00	2.37	1.40	1.65	0.57
PrintCost_Time + 2	A12	1.00	1.00	11.18	7.95	9.63	7.88	0.00	0.00	0.96	0.33	0.68	0.48
PrintCost_Time - 1	A13	1.00	1.00	20.01	23.75	17.74	16.58	0.00	0.00	1.62	1.32	1.97	0.82
AssembleCost_Time + 2	A14	1.00	1.00	7.50	7.94	7.69	7.81	0.00	0.00	0.73	0.34	0.58	0.80
AssembleCost_Time - 1	A15	1.00	1.00	22.25	23.79	20.21	16.66	0.00	0.00	1.62	1.27	1.03	0.71
Print & Assemble Time + 2	A16	1.00	1.00	4.80	3.95	4.85	3.98	0.00	0.00	0.43	0.33	0.44	0.14
BaseCost_NonPr + 1	A17	1.00	1.00	18.59	15.46	13.69	13.62	0.00	0.00	1.35	1.11	0.66	0.95
PrintCost_NonPr + 1	A18	0.99	1.00	18.50	15.15	13.62	13.77	0.10	0.00	1.34	2.00	0.56	0.60
AssembleCost_NonPr + 1	A19	1.00	1.00	18.11	15.50	13.67	13.77	0.00	0.00	1.45	0.92	0.77	0.71
[All]CostPrintable + 1	A20	1.00	1.00	11.28	13.62	9.86	9.87	0.00	0.00	0.94	0.68	0.47	0.37
[All]CostPrintable + 2	A21	1.00	1.00	6.82	10.73	6.86	7.85	0.00	0.00	0.46	0.53	0.40	0.36
Print & Assemble Pr + 2	A22	1.00	1.00	11.30	12.45	9.74	9.90	0.00	0.00	0.96	0.86	0.65	0.33
Base & Print Pr + 2	A23	1.00	1.00	8.60	12.58	7.75	8.82	0.00	0.00	0.71	0.81	0.63	0.44
Base & Assemble Pr + 2	A24	1.00	1.00	8.88	12.50	9.88	8.93	0.00	0.00	0.69	0.93	0.41	0.26
[All]CostPrintable - 1	A25	0.99	1.00	21.66	15.64	13.61	13.65	0.10	0.00	2.14	0.92	0.85	0.91
AssembleCost_Pr - 1	A26	0.98	1.00	20.98	15.65	13.59	13.70	0.14	0.00	1.42	0.78	0.82	0.73
PrintCost_Pr - 1	A27	1.00	1.00	20.72	15.54	13.63	13.74	0.00	0.00	1.99	1.67	0.61	0.54
BaseCost_Pr - 1	A28	1.00	1.00	21.40	15.54	13.68	13.69	0.00	0.00	1.96	1.11	0.58	0.83
Print_Efficiency = 0.25	B1	1.00	1.00	12.35	15.34	10.77	13.77	0.00	0.00	0.80	1.62	0.47	0.65
Print_Efficiency = 0.5	B2	1.00	1.00	15.10	15.60	12.59	13.67	0.00	0.00	1.01	0.75	0.79	0.84
Print_Efficiency = 1.5	B3	1.00	1.00	21.41	15.62	13.53	13.74	0.00	0.00	1.92	0.98	0.77	0.76
Collect_Amount = 0.25	B4	1.00	1.00	16.22	15.68	12.80	13.68	0.00	0.00	1.23	0.68	1.22	0.68
Collect_Amount = 0.5	B5	1.00	1.00	17.36	15.45	13.09	13.72	0.00	0.00	1.40	1.71	0.91	0.74
Collect_Amount = 1.5	B6	0.99	1.00	18.65	15.59	13.64	13.74	0.10	0.00	1.25	1.01	0.75	0.81
Print_Amount = 0.25	B7	1.00	1.00	12.61	15.60	10.84	13.64	0.00	0.00	0.80	0.80	0.47	0.92
Print_Amount = 0.5	B8	0.98	1.00	17.25	15.74	12.65	13.75	0.14	0.00	1.31	0.61	1.30	0.73
Print_Amount = 1.5	B9	1.00	1.00	19.55	15.48	13.16	13.43	0.00	0.00	1.99	0.99	0.94	1.17
Collect & Print Amount = 0.5	B10	1.00	1.00	16.85	15.37	12.57	13.45	0.00	0.00	1.36	1.13	0.99	1.60
QualityThreshold + 0.1	C1	1.00	1.00	17.74	15.09	13.05	13.24	0.00	0.00	1.66	1.46	1.25	1.19
QualityThreshold + 0.2	C2	1.00	1.00	16.29	14.04	12.05	12.34	0.00	0.00	2.49	1.80	1.92	1.77
QualityThreshold + 0.3	C3	1.00	1.00	13.18	11.58	9.90	10.70	0.00	0.00	3.20	2.84	2.17	2.04
QualityThreshold + 0.4	C4	0.99	1.00	9.11	8.18	6.99	7.07	0.10	0.00	3.08	2.95	1.96	2.64
RiskAmount_Print = 1%	C5	0.98	1.00	17.39	14.91	12.30	13.26	0.14	0.00	1.74	1.01	1.44	1.21
RiskAmount_Assemble = 1%	C6	0.99	1.00	17.19	13.47	11.28	12.46	0.10	0.00	1.89	2.81	2.01	1.75

(continues)

Table A.2. Print potential and standard deviation for time-step 30 (continued).

Base Algorithm	ID	Print Potential						Print Potential Std Dev					
Time-Step: 30		CHE	CHO	DHE	DHO	HHE	HHO	CHE	CHO	DHE	DHO	HHE	HHO
RiskAmount Pr & As = 10%	C7	0.79	0.96	2.43	4.26	1.80	3.75	0.41	0.20	1.58	2.11	1.12	1.99
RiskAmount Pr & As = 15%	C8	0.71	0.97	1.48	2.36	1.13	2.31	0.46	0.17	0.98	1.53	0.90	1.50
RiskAmount_Assemble = 15%	C9	0.99	1.00	3.47	4.98	3.44	4.96	0.10	0.00	1.87	2.28	1.66	1.88
Quality_incr_Chance = 0.01%	C10	0.99	1.00	18.11	15.39	13.48	13.72	0.10	0.00	1.74	1.21	1.05	0.59
Quality_decr_Chance = 25%	C11	1.00	1.00	19.10	15.72	13.75	13.93	0.00	0.00	0.98	0.73	0.66	0.29
Quality_decr_Chance = 75%	C12	1.00	1.00	17.46	15.20	13.31	13.44	0.00	0.00	1.94	1.16	0.95	0.80
Quality_decr_Upper = 0.5	C13	1.00	1.00	15.55	12.91	11.89	12.20	0.00	0.00	2.21	2.47	1.63	1.60
Qual_incr Chance & Upper * 2	C14	0.99	1.00	18.34	15.70	13.50	13.64	0.10	0.00	2.30	0.58	0.92	0.56
RiskQuality_Modifier = 10.0	C15	0.99	1.00	18.16	15.49	13.16	13.67	0.10	0.00	1.42	1.03	1.32	0.73
RiskQuality_Modifier = 25.0	C16	1.00	1.00	16.52	14.94	12.98	13.60	0.00	0.00	1.91	1.89	1.17	0.92
RiskFactory_Modifier = 0.5	C17	0.99	1.00	18.47	15.68	13.26	13.67	0.10	0.00	1.32	0.60	1.32	0.78
RiskFactory_Modifier = 1.0	C18	0.97	0.99	18.17	15.50	13.31	13.61	0.17	0.10	1.86	1.04	1.52	1.03
Quality Thres & Chance	C19	1.00	1.00	5.13	5.21	3.93	5.19	0.00	0.00	2.26	2.49	1.60	2.11
Initial_Printable / 2.0	D1	1.00	1.00	13.03	12.64	11.51	7.99	0.00	0.00	1.10	0.66	1.26	0.41
Initial_Printable * 2.0	D2	1.00	1.00	21.79	15.58	13.38	13.71	0.00	0.00	1.95	1.12	1.12	0.71
Initial_Materials = 0	D3	0.99	1.00	14.05	14.85	13.12	13.76	0.10	0.00	1.14	1.65	1.09	0.53
Initial_Materials / 2.0	D4	1.00	1.00	16.53	15.27	13.52	13.63	0.00	0.00	1.22	1.02	0.87	0.79
Initial_Materials * 2.0	D5	1.00	1.00	21.11	15.60	13.63	13.78	0.00	0.00	2.24	0.97	0.75	0.54
Env_Materials / 2.0	D6	1.00	1.00	18.78	15.46	13.56	13.76	0.00	0.00	1.25	1.07	0.74	0.74
Env_Materials * 2.0	D7	1.00	1.00	18.54	15.74	13.53	13.68	0.00	0.00	1.43	0.58	0.89	0.89
Env_Materials * 100	D8	1.00	1.00	18.59	15.49	13.45	13.53	0.00	0.00	1.36	0.95	0.86	0.96
Initial_NonPr / 2.0	D9	1.00	1.00	18.42	15.60	13.71	13.81	0.00	0.00	1.40	0.72	0.70	0.53
Initial_NonPr * 2.0	D10	1.00	1.00	18.47	15.23	13.59	13.77	0.00	0.00	1.26	1.77	0.74	0.49
Initial NonPr & Env * 2.0	D11	1.00	1.00	18.27	15.44	13.46	13.64	0.00	0.00	1.61	1.25	1.03	0.92
Initial NonPr & Env * 2.0, Raw=0	D12	1.00	1.00	14.04	15.12	13.16	13.67	0.00	0.00	1.71	1.10	1.12	0.75

Table A.3. Collection potential and standard deviation for time-step 30.

Base Algorithm	ID	Collection Potential						Collection Potential Std Dev					
Time-Step: 30		CHE	CHO	DHE	DHO	HHE	HHO	CHE	CHO	DHE	DHO	HHE	HHO
(Default)	A0	13.8	12.8	41.3	15.8	44.2	23.2	1.54	1.24	2.33	0.52	4.96	1.59
BaseCost_Pr + 1	A1	13.8	12.8	32.9	15.6	39.3	22.2	1.26	1.70	1.59	0.71	2.01	1.88
BaseCost_Pr + 3	A2	13.9	12.6	26.3	13.5	26.8	18.5	1.66	1.36	1.18	1.03	1.62	1.31
BaseCost_Pr + 5	A3	13.8	12.8	19.3	11.7	19.9	13.3	1.46	1.43	1.16	0.54	1.33	0.84
PrintCost_Pr + 1	A4	13.7	12.8	37.0	15.5	43.6	23.2	1.22	1.21	1.48	1.11	1.97	1.57
PrintCost_Pr + 3	A5	13.8	12.7	32.9	13.6	40.7	22.6	1.32	1.30	1.66	0.72	1.77	1.63
PrintCost_Pr + 5	A6	13.4	12.7	32.0	11.8	39.3	26.3	1.74	1.30	1.84	0.49	2.74	1.60
AssembleCost_Pr + 1	A7	13.7	12.5	36.5	15.5	44.0	23.4	1.37	1.37	1.68	1.05	2.59	1.32
AssembleCost_Pr + 3	A8	13.9	12.8	31.2	13.7	38.4	22.1	1.38	1.16	1.60	0.67	1.98	2.05
AssembleCost_Pr + 5	A9	13.9	12.8	26.6	11.8	34.0	25.8	1.22	1.12	1.23	0.52	1.92	1.85
BaseCost_Time + 2	A10	8.1	7.0	12.7	7.9	15.6	10.4	0.95	0.92	0.69	0.37	0.89	0.90
BaseCost_Time - 1	A11	25.9	25.4	62.7	23.9	77.8	36.5	2.85	1.72	4.44	1.39	5.39	2.50
PrintCost_Time + 2	A12	13.8	12.8	31.5	8.0	35.3	13.2	1.39	1.38	1.76	0.33	1.77	1.14
PrintCost_Time - 1	A13	13.8	12.6	46.3	23.9	54.7	29.5	1.23	1.44	2.58	1.24	6.12	1.59
AssembleCost_Time + 2	A14	13.8	12.6	19.3	7.9	21.2	13.1	1.20	1.28	1.26	0.34	1.15	1.58
AssembleCost_Time - 1	A15	13.6	12.8	54.5	23.9	66.2	29.5	1.42	1.41	2.98	1.14	3.37	1.84
Print & Assemble Time + 2	A16	13.7	12.6	12.7	4.0	17.3	6.6	1.18	1.31	0.63	0.33	1.06	0.62
BaseCost_NonPr + 1	A17	14.0	12.9	41.6	15.5	44.7	23.0	1.23	1.36	2.04	1.11	1.88	1.90
PrintCost_NonPr + 1	A18	13.6	13.0	41.6	15.2	44.8	23.3	1.75	1.18	1.83	2.00	1.69	1.59
AssembleCost_NonPr + 1	A19	13.7	12.6	41.2	15.6	44.2	23.4	1.42	1.81	2.11	0.87	2.71	1.59
[All]CostPrintable + 1	A20	13.7	12.5	29.4	13.7	35.7	21.3	1.28	1.37	1.82	0.60	2.15	1.48
[All]CostPrintable + 2	A21	13.8	12.8	22.3	10.8	24.8	19.6	1.37	1.32	1.45	0.49	1.27	1.36
Print & Assemble Pr + 2	A22	13.8	12.8	29.7	12.6	40.0	26.2	1.25	1.39	1.76	0.71	3.38	1.77
Base & Print Pr + 2	A23	13.9	12.8	27.9	12.7	30.2	21.3	1.24	1.51	1.79	0.78	1.81	1.31
Base & Assemble Pr + 2	A24	13.9	12.7	25.2	12.6	27.0	21.5	1.42	1.46	1.15	0.85	1.33	1.44
[All]CostPrintable - 1	A25	13.9	12.7	48.1	15.7	44.3	25.4	1.21	1.10	3.71	0.90	2.72	2.84
AssembleCost_Pr - 1	A26	13.8	12.8	47.0	15.7	44.5	22.9	1.19	1.37	2.73	0.78	2.57	1.75
PrintCost_Pr - 1	A27	13.7	12.8	45.9	15.6	44.5	23.4	1.59	1.21	3.35	1.67	2.22	1.27
BaseCost_Pr - 1	A28	13.6	12.7	47.6	15.6	44.5	26.5	1.48	1.29	3.04	1.08	1.94	2.53
Print_Efficiency = 0.25	B1	13.7	12.7	29.0	15.4	35.8	21.5	1.33	1.31	1.26	1.61	2.16	1.42
Print_Efficiency = 0.5	B2	13.8	12.8	33.1	15.7	40.3	23.4	1.25	1.61	1.48	0.69	2.02	1.63
Print_Efficiency = 1.5	B3	13.9	12.7	47.6	15.7	44.2	25.0	1.38	1.31	2.94	0.97	2.64	2.04
Collect_Amount = 0.25	B4	13.8	12.8	37.8	15.7	42.5	23.1	1.25	1.41	1.93	0.67	3.04	1.67
Collect_Amount = 0.5	B5	13.8	12.6	39.4	15.5	43.8	23.3	1.43	1.35	2.23	1.70	2.18	1.53
Collect_Amount = 1.5	B6	13.6	13.0	42.5	15.7	44.4	23.3	1.36	1.18	2.32	0.93	2.10	1.52
Print_Amount = 0.25	B7	13.7	13.0	31.2	15.7	36.3	21.6	1.30	1.25	1.53	0.78	1.45	1.39
Print_Amount = 0.5	B8	14.0	12.7	37.2	15.8	40.0	23.6	1.25	1.22	1.86	0.54	4.12	1.34
Print_Amount = 1.5	B9	13.6	12.9	43.9	15.5	43.9	24.5	1.58	1.25	2.77	0.99	2.43	2.65
Collect & Print Amount = 0.5	B10	13.8	12.6	36.9	15.6	39.9	22.9	1.20	1.82	1.99	1.05	2.69	2.98
QualityThreshold + 0.1	C1	13.5	12.8	40.1	15.2	42.2	22.5	1.38	1.35	2.55	1.46	3.58	2.09
QualityThreshold + 0.2	C2	13.7	12.9	38.0	14.1	39.4	20.6	1.27	1.17	4.02	1.80	5.74	3.13
QualityThreshold + 0.3	C3	12.8	11.9	31.0	11.7	31.0	17.7	1.44	1.41	6.81	2.87	6.96	3.55
QualityThreshold + 0.4	C4	10.7	9.3	21.2	8.2	21.8	11.1	1.84	1.88	6.98	2.94	6.47	4.76
RiskAmount_Print = 1%	C5	13.8	12.8	41.4	15.6	44.0	22.9	1.30	1.34	2.17	0.63	2.47	1.90
RiskAmount_Assemble = 1%	C6	13.6	12.5	39.2	13.5	37.3	21.4	1.70	1.73	3.52	2.84	6.64	3.36

(continues)

Table A.3. Collection potential and standard deviation for time-step 30 (continued).

Base Algorithm	ID	Collection Potential						Collection Potential Std Dev					
Time-Step: 30		CHE	CHO	DHE	DHO	HHE	HHO	CHE	CHO	DHE	DHO	HHE	HHO
RiskAmount Pr & As = 10%	C7	12.2	11.1	14.2	6.8	13.9	8.7	3.61	3.52	6.00	2.40	5.89	4.01
RiskAmount Pr & As = 15%	C8	11.3	10.5	10.9	5.0	10.7	7.0	4.21	3.78	4.48	2.46	5.05	3.18
RiskAmount_Assemble = 15%	C9	11.9	10.4	10.7	5.0	11.5	7.0	3.62	3.85	4.81	2.27	4.62	3.08
Quality_incr_Chance = 0.01%	C10	13.9	12.7	40.8	15.5	43.9	23.1	1.31	1.38	3.25	1.22	3.30	1.37
Quality_decr_Chance = 25%	C11	13.9	13.0	42.6	15.8	45.2	23.6	1.31	1.21	1.26	0.70	2.16	1.11
Quality_decr_Chance = 75%	C12	13.4	12.3	39.7	15.3	42.9	22.3	1.36	1.55	3.20	1.15	3.12	1.80
Quality_decr_Upper = 0.5	C13	13.2	12.6	35.7	13.0	38.1	20.1	1.45	1.37	4.00	2.50	4.79	3.25
Qual_incr Chance & Upper * 2	C14	13.7	12.8	41.2	15.7	44.4	23.1	1.38	1.14	4.48	0.54	2.69	1.53
RiskQuality_Modifier = 10.0	C15	13.2	12.3	41.1	15.6	42.6	22.8	1.49	1.31	2.45	1.00	3.94	1.62
RiskQuality_Modifier = 25.0	C16	12.4	11.0	38.4	15.2	41.4	21.6	1.34	1.66	2.85	1.87	3.75	2.22
RiskFactory_Modifier = 0.5	C17	13.8	12.7	41.7	15.7	43.4	23.2	1.58	1.52	1.91	0.57	3.86	1.54
RiskFactory_Modifier = 1.0	C18	13.4	12.3	40.9	15.5	43.4	22.9	1.85	2.11	3.42	1.00	5.10	2.00
Quality Thres & Chance	C19	8.7	7.2	12.6	5.2	12.0	7.2	1.90	2.11	5.01	2.49	4.82	3.36
Initial_Printable / 2.0	D1	13.7	12.9	31.9	12.8	38.2	24.9	1.22	1.30	1.81	0.58	2.21	1.75
Initial_Printable * 2.0	D2	13.8	12.9	48.0	15.6	44.0	25.8	1.26	1.14	3.34	1.12	2.70	2.09
Initial_Materials = 0	D3	13.7	12.6	34.2	14.9	43.0	23.4	1.39	1.68	2.20	1.65	3.07	1.43
Initial_Materials / 2.0	D4	13.7	12.9	37.7	15.3	44.2	23.1	1.66	1.41	1.76	1.00	2.80	1.68
Initial_Materials * 2.0	D5	13.5	12.8	47.3	15.7	44.3	23.4	1.38	1.36	4.49	0.97	1.87	1.41
Env_Materials / 2.0	D6	13.5	12.6	41.9	15.5	44.2	23.3	1.57	1.40	1.81	1.01	2.55	1.74
Env_Materials * 2.0	D7	13.9	12.5	41.5	15.8	44.3	23.1	1.41	1.47	2.05	0.56	2.65	2.06
Env_Materials * 100	D8	13.7	12.8	41.6	15.6	44.0	23.1	1.32	1.26	2.03	0.91	2.59	1.85
Initial_NonPr / 2.0	D9	13.6	12.6	41.4	15.7	44.7	23.3	1.43	1.34	2.23	0.67	2.16	1.39
Initial_NonPr * 2.0	D10	13.7	12.7	41.4	15.4	44.1	23.3	1.37	1.36	2.06	1.76	2.54	1.31
Initial NonPr & Env * 2.0	D11	13.7	13.0	41.1	15.5	44.1	23.0	1.45	1.32	2.62	1.23	3.16	2.03
Initial NonPr & Env * 2.0, Raw=0	D12	13.7	12.6	33.7	15.2	43.4	23.1	1.45	1.31	3.75	1.08	2.64	1.66

A.2. Base Algorithm: Time-Step 50

In this subsection, the assembly, print, and collection potentials are provided for time-step 50. Table A.4 lists the data for assembly potential. Table A.5 lists the data for print potential. Table A.6 lists the data for the collection potential.

Table A.4. Assembly potential and standard deviation for time-step 50.

Base Algorithm	ID	Assembly Potential						Assembly Potential Std Dev					
Time-Step: 50		CHE	CHO	DHE	DHO	HHE	HHO	CHE	CHO	DHE	DHO	HHE	HHO
(Default)	A0	1.00	1.00	69.88	38.93	63.27	32.19	0.00	0.00	6.60	3.35	3.30	1.43
BaseCost_Pr + 1	A1	1.00	0.99	52.12	33.11	45.17	25.55	0.00	0.10	7.46	2.91	3.72	1.01
BaseCost_Pr + 3	A2	1.00	0.99	31.98	24.54	28.89	17.52	0.00	0.10	2.88	2.36	4.14	1.10
BaseCost_Pr + 5	A3	1.00	1.00	22.38	19.83	15.54	16.92	0.00	0.00	2.51	2.13	2.02	2.09
PrintCost_Pr + 1	A4	1.00	1.00	63.96	32.79	58.35	28.63	0.00	0.00	5.30	2.74	4.67	2.36
PrintCost_Pr + 3	A5	1.00	1.00	52.84	25.08	47.18	11.81	0.00	0.00	5.77	2.50	3.47	1.34
PrintCost_Pr + 5	A6	0.99	0.99	53.66	19.64	51.73	8.76	0.10	0.10	7.09	2.20	4.23	0.49
AssembleCost_Pr + 1	A7	1.00	0.99	57.25	33.15	57.76	28.25	0.00	0.10	5.71	2.99	4.10	2.45
AssembleCost_Pr + 3	A8	1.00	0.98	41.04	24.50	38.25	11.60	0.00	0.14	3.24	2.02	2.74	1.04
AssembleCost_Pr + 5	A9	0.99	1.00	30.11	19.04	29.67	8.71	0.10	0.00	2.54	2.81	1.86	0.62
BaseCost_Time + 2	A10	1.00	1.00	30.64	24.56	23.13	18.06	0.00	0.00	2.17	1.68	1.29	1.52
BaseCost_Time - 1	A11	0.98	1.00	73.89	51.20	62.35	50.58	0.14	0.00	4.69	4.81	2.79	3.62
PrintCost_Time + 2	A12	1.00	1.00	57.64	24.69	53.63	21.34	0.00	0.00	4.08	2.03	3.44	1.30
PrintCost_Time - 1	A13	0.99	1.00	75.01	50.23	65.24	43.76	0.10	0.00	3.45	4.87	2.54	2.78
AssembleCost_Time + 2	A14	0.99	1.00	38.83	24.68	36.14	21.25	0.10	0.00	3.62	1.73	2.87	1.37
AssembleCost_Time - 1	A15	1.00	0.99	75.31	51.37	63.69	43.70	0.00	0.10	3.96	4.41	2.75	4.95
Print & Assemble Time + 2	A16	0.98	1.00	30.34	15.19	27.66	14.33	0.14	0.00	1.96	1.13	1.77	1.41
BaseCost_NonPr + 1	A17	1.00	1.00	48.41	38.74	39.87	31.88	0.00	0.00	3.53	3.21	2.65	1.85
PrintCost_NonPr + 1	A18	1.00	1.00	59.91	39.61	53.63	31.92	0.00	0.00	3.82	2.97	2.85	1.90
AssembleCost_NonPr + 1	A19	1.00	0.99	57.15	39.61	52.32	32.05	0.00	0.10	3.61	2.96	2.43	1.65
[All]CostPrintable + 1	A20	0.98	1.00	41.39	24.56	33.05	15.89	0.14	0.00	4.06	2.49	2.24	2.25
[All]CostPrintable + 2	A21	0.99	1.00	27.14	17.54	22.52	7.67	0.10	0.00	2.73	1.76	3.26	0.95
Print & Assemble Pr + 2	A22	1.00	1.00	40.77	22.25	34.37	9.81	0.00	0.00	4.32	2.01	1.93	0.44
Base & Print Pr + 2	A23	1.00	0.98	39.88	21.85	30.51	8.98	0.00	0.14	4.11	2.19	4.65	0.86
Base & Assemble Pr + 2	A24	1.00	1.00	30.57	22.40	27.98	9.01	0.00	0.00	3.07	2.07	2.39	0.96
[All]CostPrintable - 1	A25	1.00	0.99	69.99	77.90	62.35	52.25	0.00	0.10	4.65	5.31	2.71	3.42
AssembleCost_Pr - 1	A26	1.00	0.99	74.59	48.18	63.70	38.62	0.00	0.10	4.50	3.51	3.00	3.06
PrintCost_Pr - 1	A27	0.99	0.99	71.38	47.69	62.89	38.53	0.10	0.10	4.70	3.74	2.74	2.84
BaseCost_Pr - 1	A28	1.00	1.00	71.61	47.08	62.36	37.12	0.00	0.00	4.49	4.14	2.72	3.34
Print_Efficiency = 0.25	B1	0.99	1.00	25.73	20.60	18.29	14.60	0.10	0.00	2.62	1.90	2.24	0.74
Print_Efficiency = 0.5	B2	1.00	0.99	37.07	26.67	33.97	24.08	0.00	0.10	4.00	2.43	2.47	1.29
Print_Efficiency = 1.5	B3	1.00	0.99	72.48	49.80	62.20	38.21	0.00	0.10	4.05	4.40	2.70	2.56
Collect_Amount = 0.25	B4	0.98	1.00	35.77	27.58	33.34	27.21	0.14	0.00	3.22	2.59	2.54	2.40
Collect_Amount = 0.5	B5	0.99	1.00	50.36	32.61	49.53	31.55	0.10	0.00	5.04	3.08	4.02	2.35
Collect_Amount = 1.5	B6	0.98	0.99	75.80	42.60	64.88	31.64	0.14	0.10	3.99	3.63	2.70	2.16
Print_Amount = 0.25	B7	1.00	0.99	27.81	27.26	18.66	14.64	0.00	0.10	2.99	1.77	2.42	0.70

(continues)

Table A.4. Assembly potential and standard deviation for time-step 50 (continued).

Base Algorithm	ID	Assembly Potential						Assembly Potential Std Dev					
Time-Step: 50		CHE	CHO	DHE	DHO	HHE	HHO	CHE	CHO	DHE	DHO	HHE	HHO
Print_Amount = 0.5	B8	0.99	0.99	44.10	32.49	33.95	23.36	0.10	0.10	4.15	2.22	2.74	2.55
Print_Amount = 1.5	B9	0.99	1.00	74.41	42.04	63.64	37.51	0.10	0.00	3.88	3.92	3.11	3.29
Collect & Print Amount = 0.5	B10	1.00	1.00	40.04	29.34	32.11	23.57	0.00	0.00	3.70	2.09	2.45	2.35
QualityThreshold + 0.1	C1	0.99	1.00	66.29	37.49	60.85	31.02	0.10	0.00	8.02	4.04	4.79	2.63
QualityThreshold + 0.2	C2	1.00	0.99	57.88	35.85	52.51	28.83	0.00	0.10	9.53	4.71	7.02	3.86
QualityThreshold + 0.3	C3	0.99	1.00	45.29	28.36	43.34	23.58	0.10	0.00	9.05	5.56	8.24	4.45
QualityThreshold + 0.4	C4	0.99	0.99	30.19	20.60	26.32	15.85	0.10	0.10	6.65	4.89	6.62	5.12
RiskAmount_Print = 1%	C5	1.00	0.99	68.48	38.90	58.65	30.95	0.00	0.10	6.43	3.55	5.73	1.92
RiskAmount_Assemble = 1%	C6	0.97	0.97	54.06	32.56	52.65	26.35	0.17	0.17	11.1	5.78	6.69	4.00
RiskAmount Pr & As = 10%	C7	0.64	0.63	5.49	4.50	3.58	2.57	0.48	0.49	4.70	4.17	2.85	2.14
RiskAmount Pr & As = 15%	C8	0.48	0.45	1.77	1.72	1.84	1.40	0.50	0.50	2.15	2.07	1.94	1.41
RiskAmount_Assemble = 15%	C9	0.43	0.49	1.71	1.85	1.43	1.24	0.50	0.50	1.66	2.02	1.58	1.34
Quality_incr_Chance = 0.01%	C10	1.00	1.00	69.83	38.48	62.74	31.86	0.00	0.00	5.56	3.51	3.72	2.01
Quality_decr_Chance = 25%	C11	1.00	1.00	74.03	40.62	65.22	32.14	0.00	0.00	4.52	2.60	3.12	1.65
Quality_decr_Chance = 75%	C12	0.99	1.00	62.91	36.91	59.84	30.66	0.10	0.00	8.96	4.05	4.07	4.08
Quality_decr_Upper = 0.5	C13	1.00	1.00	52.25	31.95	49.02	27.53	0.00	0.00	8.87	5.51	8.18	3.66
Qual_incr Chance & Upper * 2	C14	1.00	0.99	70.33	39.33	63.65	32.05	0.00	0.10	7.00	2.94	3.08	3.47
RiskQuality_Modifier = 10.0	C15	1.00	1.00	65.42	37.58	61.49	31.66	0.00	0.00	7.06	3.44	4.58	2.11
RiskQuality_Modifier = 25.0	C16	0.99	1.00	54.54	33.84	55.41	30.69	0.10	0.00	8.49	4.28	6.26	2.74
RiskFactory_Modifier = 0.5	C17	0.96	0.98	68.63	39.05	61.81	31.74	0.20	0.14	7.09	3.14	7.39	2.36
RiskFactory_Modifier = 1.0	C18	0.93	0.93	68.16	38.36	62.37	31.34	0.26	0.26	7.33	4.95	7.13	3.82
Quality Thres & Chance	C19	1.00	1.00	15.80	12.27	14.51	9.34	0.00	0.00	4.78	4.13	4.94	3.53
Initial_Printable / 2.0	D1	0.99	0.99	57.29	28.05	51.90	8.22	0.10	0.10	5.16	2.43	4.00	1.74
Initial_Printable * 2.0	D2	1.00	1.00	73.05	56.46	63.48	44.35	0.00	0.00	3.69	4.74	2.45	2.23
Initial_Materials = 0	D3	0.99	0.99	61.08	32.42	59.32	31.91	0.10	0.10	6.81	3.36	5.09	1.71
Initial_Materials / 2.0	D4	0.99	1.00	65.55	36.50	63.05	32.02	0.10	0.00	8.87	3.26	3.70	1.48
Initial_Materials * 2.0	D5	1.00	0.99	73.27	43.03	63.11	31.78	0.00	0.10	6.26	5.65	2.92	3.52
Env_Materials / 2.0	D6	1.00	0.99	52.84	39.20	44.59	31.97	0.00	0.10	2.67	3.15	1.94	1.64
Env_Materials * 2.0	D7	0.99	1.00	67.21	39.73	60.22	31.70	0.10	0.00	6.36	3.11	3.37	3.68
Env_Materials * 100	D8	0.99	1.00	68.45	38.91	60.15	31.77	0.10	0.00	6.19	3.00	3.47	1.97
Initial_NonPr / 2.0	D9	1.00	1.00	34.72	39.17	32.11	31.96	0.00	0.00	3.19	3.10	2.26	1.95
Initial_NonPr * 2.0	D10	0.99	1.00	70.76	39.13	66.70	31.76	0.10	0.00	6.31	5.09	3.35	2.02
Initial NonPr & Env * 2.0	D11	1.00	1.00	68.11	39.04	64.63	32.03	0.00	0.00	6.51	3.16	4.73	1.43
Initial NonPr & Env * 2.0, Raw=0	D12	1.00	1.00	61.34	33.20	57.47	31.98	0.00	0.00	6.11	3.29	4.98	1.59

Table A.5. Print potential and standard deviation for time-step 50.

Base Algorithm	ID	Print Potential						Print Potential Std Dev					
Time-Step: 50		CHE	CHO	DHE	DHO	HHE	HHO	CHE	CHO	DHE	DHO	HHE	HHO
(Default)	A0	0.98	1.00	58.22	38.81	53.07	32.19	0.14	0.00	6.01	3.14	2.93	1.40
BaseCost_Pr + 1	A1	1.00	1.00	41.00	33.12	36.35	25.54	0.00	0.00	5.46	2.59	2.97	1.00
BaseCost_Pr + 3	A2	1.00	1.00	20.66	24.50	6.80	17.36	0.00	0.00	4.53	2.22	0.57	1.27
BaseCost_Pr + 5	A3	0.99	1.00	5.60	19.70	4.79	16.77	0.10	0.00	0.67	1.93	0.43	2.04
PrintCost_Pr + 1	A4	0.99	1.00	49.52	32.74	48.30	28.70	0.10	0.00	4.12	2.58	4.62	2.22
PrintCost_Pr + 3	A5	1.00	1.00	36.22	24.88	35.05	11.92	0.00	0.00	3.19	2.55	4.78	1.34
PrintCost_Pr + 5	A6	1.00	1.00	17.49	19.49	10.56	8.81	0.00	0.00	6.01	2.25	2.16	0.39
AssembleCost_Pr + 1	A7	0.99	1.00	49.53	32.98	51.31	28.16	0.10	0.00	4.77	2.90	4.48	2.32
AssembleCost_Pr + 3	A8	1.00	1.00	40.86	24.50	33.49	11.69	0.00	0.00	3.89	2.02	4.82	1.14
AssembleCost_Pr + 5	A9	1.00	1.00	26.67	19.04	27.31	8.76	0.00	0.00	4.61	2.74	6.41	0.47
BaseCost_Time + 2	A10	1.00	1.00	23.54	24.79	16.91	18.37	0.00	0.00	2.12	1.36	1.22	1.23
BaseCost_Time - 1	A11	1.00	1.00	64.28	51.08	54.10	50.43	0.00	0.00	3.19	4.62	1.73	3.67
PrintCost_Time + 2	A12	1.00	1.00	39.69	24.82	37.87	21.36	0.00	0.00	2.49	1.82	2.15	0.99
PrintCost_Time - 1	A13	1.00	1.00	62.06	50.32	53.47	43.54	0.00	0.00	3.33	4.59	2.25	2.79
AssembleCost_Time + 2	A14	0.99	1.00	30.94	24.74	32.69	21.27	0.10	0.00	3.27	1.56	3.07	1.21
AssembleCost_Time - 1	A15	1.00	1.00	63.42	51.36	54.14	43.62	0.00	0.00	2.37	4.42	1.75	4.82
Print & Assemble Time + 2	A16	1.00	1.00	23.34	15.38	22.04	14.47	0.00	0.00	1.88	0.91	2.07	1.09
BaseCost_NonPr + 1	A17	1.00	1.00	40.93	38.92	33.48	31.84	0.00	0.00	2.57	3.04	1.62	1.69
PrintCost_NonPr + 1	A18	0.99	1.00	50.00	39.55	45.09	31.96	0.10	0.00	2.94	2.78	1.88	1.83
AssembleCost_NonPr + 1	A19	0.99	1.00	48.55	39.55	43.48	32.03	0.10	0.00	2.81	2.89	1.91	1.73
[All]CostPrintable + 1	A20	1.00	1.00	30.66	24.50	19.94	15.85	0.00	0.00	2.81	2.36	5.99	2.28
[All]CostPrintable + 2	A21	1.00	1.00	6.72	17.45	6.64	7.59	0.00	0.00	0.55	1.67	0.63	0.85
Print & Assemble Pr + 2	A22	1.00	1.00	30.00	22.25	10.37	9.82	0.00	0.00	3.56	1.86	3.90	0.41
Base & Print Pr + 2	A23	1.00	1.00	9.87	21.81	7.33	8.99	0.00	0.00	3.00	2.05	1.06	0.78
Base & Assemble Pr + 2	A24	1.00	1.00	15.07	22.37	10.47	9.02	0.00	0.00	3.65	1.92	2.63	0.92
[All]CostPrintable - 1	A25	1.00	1.00	64.58	78.38	55.49	52.67	0.00	0.00	4.14	5.01	2.60	3.11
AssembleCost_Pr - 1	A26	1.00	1.00	61.57	48.13	54.40	38.68	0.00	0.00	3.44	3.32	2.04	3.00
PrintCost_Pr - 1	A27	1.00	1.00	64.09	47.85	55.89	38.49	0.00	0.00	3.38	3.59	2.48	2.76
BaseCost_Pr - 1	A28	1.00	1.00	63.57	47.02	55.88	37.28	0.00	0.00	3.45	3.83	1.93	3.21
Print_Efficiency = 0.25	B1	1.00	1.00	11.82	20.39	10.47	14.43	0.00	0.00	1.10	1.79	0.67	0.76
Print_Efficiency = 0.5	B2	1.00	1.00	31.04	26.57	17.50	23.95	0.00	0.00	2.70	2.37	4.57	1.37
Print_Efficiency = 1.5	B3	1.00	1.00	63.38	49.98	55.54	38.48	0.00	0.00	3.65	4.22	2.10	2.24
Collect_Amount = 0.25	B4	1.00	1.00	25.67	27.66	26.15	27.19	0.00	0.00	2.99	2.42	2.50	2.31
Collect_Amount = 0.5	B5	1.00	1.00	37.97	32.72	38.59	31.68	0.00	0.00	3.78	2.91	2.86	2.09
Collect_Amount = 1.5	B6	1.00	1.00	63.85	42.69	54.01	31.72	0.00	0.00	3.70	3.28	2.19	2.02
Print_Amount = 0.25	B7	1.00	1.00	12.27	27.01	10.49	14.46	0.00	0.00	0.92	1.68	0.72	0.77
Print_Amount = 0.5	B8	0.99	1.00	38.76	32.30	17.38	23.31	0.10	0.00	4.29	2.11	4.41	2.53
Print_Amount = 1.5	B9	1.00	1.00	62.40	42.32	52.90	37.69	0.00	0.00	4.03	3.61	2.97	3.21
Collect & Print Amount = 0.5	B10	0.97	1.00	33.17	29.27	17.92	23.51	0.17	0.00	2.41	2.10	4.13	2.05
QualityThreshold + 0.1	C1	0.99	1.00	55.16	37.54	50.64	30.94	0.10	0.00	7.04	3.95	4.40	2.60
QualityThreshold + 0.2	C2	0.98	1.00	47.75	36.05	43.01	28.81	0.14	0.00	8.78	4.63	7.65	3.74
QualityThreshold + 0.3	C3	1.00	1.00	35.39	28.34	33.83	23.57	0.00	0.00	7.21	5.46	8.10	4.40
QualityThreshold + 0.4	C4	1.00	1.00	22.41	20.67	19.86	15.99	0.00	0.00	6.26	4.94	5.40	5.09
RiskAmount_Print = 1%	C5	0.95	1.00	50.03	36.46	41.12	27.90	0.22	0.00	5.80	3.47	6.92	2.52
RiskAmount_Assemble = 1%	C6	1.00	1.00	49.10	35.31	49.28	29.57	0.00	0.00	9.06	5.04	5.58	2.77

(continues)

Table A.5. Print potential and standard deviation for time-step 50 (continued).

Base Algorithm	ID	Print Potential						Print Potential Std Dev					
Time-Step: 50		CHE	CHO	DHE	DHO	HHE	HHO	CHE	CHO	DHE	DHO	HHE	HHO
RiskAmount Pr & As = 10%	C7	0.52	0.90	1.81	4.14	1.42	3.94	0.50	0.30	1.52	2.80	1.25	2.57
RiskAmount Pr & As = 15%	C8	0.50	0.79	0.93	2.30	0.88	2.40	0.50	0.41	0.90	1.82	0.84	1.69
RiskAmount_Assemble = 15%	C9	0.99	1.00	6.12	8.95	4.91	7.68	0.10	0.00	3.90	4.95	3.09	3.81
Quality_incr_Chance = 0.01%	C10	1.00	1.00	58.36	38.63	53.36	31.82	0.00	0.00	4.51	3.30	2.83	1.88
Quality_decr_Chance = 25%	C11	1.00	1.00	61.76	40.69	54.32	32.24	0.00	0.00	4.04	2.42	2.70	1.53
Quality_decr_Chance = 75%	C12	1.00	1.00	52.32	37.01	50.96	30.56	0.00	0.00	7.78	3.93	3.58	3.90
Quality_decr_Upper = 0.5	C13	0.99	1.00	42.44	31.96	40.41	27.59	0.10	0.00	7.88	5.38	7.77	3.63
Qual_incr Chance & Upper * 2	C14	1.00	1.00	59.22	39.28	53.16	31.96	0.00	0.00	5.25	2.67	2.55	3.36
RiskQuality_Modifier = 10.0	C15	1.00	1.00	55.38	37.55	52.01	31.65	0.00	0.00	5.67	3.30	3.54	1.71
RiskQuality_Modifier = 25.0	C16	0.99	1.00	46.53	33.92	48.34	30.92	0.10	0.00	7.83	4.06	5.43	2.43
RiskFactory_Modifier = 0.5	C17	1.00	1.00	57.73	39.01	52.53	31.79	0.00	0.00	5.92	3.05	6.16	2.35
RiskFactory_Modifier = 1.0	C18	0.98	1.00	57.08	38.50	52.65	31.34	0.14	0.00	5.94	4.76	6.03	3.63
Quality Thres & Chance	C19	1.00	1.00	11.33	12.34	9.40	9.38	0.00	0.00	3.77	4.05	4.18	3.60
Initial_Printable / 2.0	D1	1.00	1.00	44.55	28.04	42.35	8.24	0.00	0.00	4.23	2.16	4.24	1.92
Initial_Printable * 2.0	D2	1.00	1.00	64.13	56.55	55.55	44.55	0.00	0.00	3.15	4.42	2.21	2.18
Initial_Materials = 0	D3	0.99	1.00	48.43	32.46	48.97	31.91	0.10	0.00	5.82	3.30	4.88	1.49
Initial_Materials / 2.0	D4	1.00	1.00	54.61	36.58	52.18	31.96	0.00	0.00	7.45	3.08	3.54	1.51
Initial_Materials * 2.0	D5	0.98	1.00	61.94	42.94	53.60	31.78	0.14	0.00	5.07	5.51	2.22	3.38
Env_Materials / 2.0	D6	1.00	1.00	42.11	39.17	35.76	31.86	0.00	0.00	2.23	3.00	1.32	1.54
Env_Materials * 2.0	D7	1.00	1.00	57.11	39.68	53.02	31.73	0.00	0.00	5.75	3.03	2.87	3.51
Env_Materials * 100	D8	1.00	1.00	58.09	38.94	53.02	31.73	0.00	0.00	6.22	2.81	2.89	1.81
Initial_NonPr / 2.0	D9	1.00	1.00	30.67	39.04	26.56	31.95	0.00	0.00	2.06	2.94	1.54	1.53
Initial_NonPr * 2.0	D10	0.99	1.00	58.95	39.00	56.57	31.75	0.10	0.00	5.21	4.93	3.97	1.97
Initial NonPr & Env * 2.0	D11	1.00	1.00	57.54	39.17	56.53	32.01	0.00	0.00	5.19	2.98	5.22	1.39
Initial NonPr & Env * 2.0, Raw=0	D12	1.00	1.00	49.27	33.36	48.08	32.04	0.00	0.00	5.03	3.14	4.70	1.55

Table A.6. Collection potential and standard deviation for time-step 50.

Base Algorithm	ID	Collection Potential						Collection Potential Std Dev					
Time-Step: 50		CHE	CHO	DHE	DHO	HHE	HHO	CHE	CHO	DHE	DHO	HHE	HHO
(Default)	A0	20.3	18.8	129.5	39.3	164.5	66.5	2.07	1.98	11.6	3.15	7.92	3.83
BaseCost_Pr + 1	A1	19.9	19.0	94.5	33.4	114.9	53.7	2.00	1.83	12.3	2.66	8.97	4.17
BaseCost_Pr + 3	A2	19.7	15.9	53.3	24.7	40.9	32.9	1.83	2.44	5.97	2.27	2.84	2.34
BaseCost_Pr + 5	A3	17.2	12.2	28.4	19.9	26.9	27.2	1.90	1.73	2.57	1.99	2.52	1.84
PrintCost_Pr + 1	A4	20.1	19.2	114.7	33.1	157.0	62.0	2.19	2.18	8.30	2.54	11.6	4.49
PrintCost_Pr + 3	A5	19.7	19.3	90.1	25.2	122.6	72.4	2.14	2.16	8.39	2.44	9.11	5.59
PrintCost_Pr + 5	A6	19.9	19.2	71.9	19.7	97.1	63.1	2.11	2.58	5.00	2.14	5.12	4.35
AssembleCost_Pr + 1	A7	20.0	18.9	108.1	33.3	155.7	61.1	2.00	2.46	9.52	2.83	10.1	5.04
AssembleCost_Pr + 3	A8	20.0	18.5	83.0	24.8	103.3	71.9	2.24	2.94	6.40	1.92	10.2	5.24
AssembleCost_Pr + 5	A9	20.0	18.8	57.5	19.2	80.8	62.4	2.29	2.07	5.44	2.70	7.47	4.60
BaseCost_Time + 2	A10	11.3	10.0	54.8	25.0	53.7	30.3	1.41	1.50	3.67	1.42	3.45	2.27
BaseCost_Time - 1	A11	43.5	43.1	139.5	51.6	166.5	109	3.78	2.64	6.60	4.43	6.44	7.95
PrintCost_Time + 2	A12	20.4	19.0	98.5	25.0	136.0	38.4	2.11	2.16	5.79	1.78	8.60	2.74
PrintCost_Time - 1	A13	19.9	18.9	138.6	50.8	163.9	90.8	2.16	2.06	5.03	4.55	5.63	6.68
AssembleCost_Time + 2	A14	19.8	19.2	70.8	24.9	96.8	38.9	2.41	1.96	6.15	1.51	8.60	2.41
AssembleCost_Time - 1	A15	20.3	18.8	140.4	51.8	167.7	89.4	1.80	2.44	4.87	4.21	4.81	11.00
Print & Assemble Time + 2	A16	19.4	18.8	54.3	15.4	70.1	27.6	2.27	2.06	3.19	0.82	5.00	2.54
BaseCost_NonPr + 1	A17	20.0	19.1	90.6	39.2	100.3	65.5	1.87	2.14	4.69	3.06	4.99	4.00
PrintCost_NonPr + 1	A18	20.3	19.3	111.1	39.9	138.6	65.8	1.78	2.08	5.74	2.78	5.72	3.78
AssembleCost_NonPr + 1	A19	20.0	18.9	107.0	39.9	133.4	66.5	2.33	2.57	5.15	2.71	4.84	4.81
[All]CostPrintable + 1	A20	19.9	18.6	73.0	24.8	75.8	42.8	2.28	2.15	6.20	2.35	6.57	3.49
[All]CostPrintable + 2	A21	19.7	18.8	34.3	17.6	38.9	38.5	2.65	1.94	2.74	1.70	2.81	5.01
Print & Assemble Pr + 2	A22	20.3	18.8	71.8	22.4	88.8	67.1	2.25	2.09	7.01	1.87	7.65	4.92
Base & Print Pr + 2	A23	20.4	18.8	50.3	22.0	50.3	43.7	2.18	2.34	3.66	2.03	6.90	3.44
Base & Assemble Pr + 2	A24	19.5	19.2	46.2	22.6	49.1	43.7	2.21	2.15	5.15	1.86	3.92	3.86
[All]CostPrintable - 1	A25	20.1	19.2	136.1	78.8	162.9	112	1.90	2.52	7.25	4.90	7.82	7.28
AssembleCost_Pr - 1	A26	19.7	18.9	137.6	48.6	164.0	72.8	1.91	1.82	6.57	3.29	6.49	5.02
PrintCost_Pr - 1	A27	19.7	19.1	137.1	48.2	164.0	72.3	2.61	2.80	6.52	3.57	6.28	4.70
BaseCost_Pr - 1	A28	19.9	18.7	136.7	47.5	163.3	87.9	1.76	2.03	6.45	3.83	6.25	6.40
Print_Efficiency = 0.25	B1	19.7	19.1	38.2	20.7	49.5	45.0	2.49	2.27	2.88	1.80	3.53	3.19
Print_Efficiency = 0.5	B2	20.1	19.0	69.0	26.9	71.9	44.7	2.08	2.66	6.08	2.28	4.27	3.00
Print_Efficiency = 1.5	B3	19.9	18.9	137.5	50.3	163.0	81.3	1.97	2.58	6.21	4.10	6.47	5.39
Collect_Amount = 0.25	B4	19.8	19.1	61.8	27.8	80.9	47.8	2.48	2.08	5.17	2.42	7.80	4.87
Collect_Amount = 0.5	B5	19.7	19.2	89.2	32.9	127.1	64.9	2.51	1.93	7.71	2.88	10.7	5.97
Collect_Amount = 1.5	B6	21.2	20.2	141.4	43.1	168.8	67.8	3.31	2.41	6.45	3.24	5.53	4.69
Print_Amount = 0.25	B7	20.1	18.9	40.6	27.4	49.3	45.5	2.12	2.29	3.27	1.58	3.60	3.30
Print_Amount = 0.5	B8	19.7	19.1	84.1	32.8	70.7	44.2	2.07	2.10	7.65	2.04	4.81	3.41
Print_Amount = 1.5	B9	19.7	18.7	138.1	42.5	164.2	80.3	2.52	1.67	6.81	3.62	7.11	6.59
Collect & Print Amount = 0.5	B10	19.9	19.0	74.4	29.6	67.6	44.7	1.98	2.03	5.03	2.00	4.53	3.10
QualityThreshold + 0.1	C1	19.9	19.2	122.9	37.8	158.0	63.2	2.53	2.24	14.0	3.88	11.0	5.84
QualityThreshold + 0.2	C2	19.7	18.9	106.9	36.3	136.9	58.4	2.05	2.62	17.5	4.60	18.3	8.36
QualityThreshold + 0.3	C3	18.0	17.4	81.6	28.6	109.4	45.4	2.70	2.33	15.5	5.47	22.9	9.42
QualityThreshold + 0.4	C4	15.0	14.8	53.2	20.8	64.2	30.7	2.58	2.32	12.1	4.93	16.3	9.35
RiskAmount_Print = 1%	C5	20.3	18.9	127.4	39.3	154.2	63.6	2.23	2.50	11.0	3.23	14.8	4.93
RiskAmount_Assemble = 1%	C6	19.6	18.5	110.3	35.7	153.3	61.7	2.95	2.94	19.6	5.07	14.3	6.26

(continues)

Table A.6. Collection potential and standard deviation for time-step 50 (continued).

Base Algorithm	ID	Collection Potential						Collection Potential Std Dev					
Time-Step: 50		CHE	CHO	DHE	DHO	HHE	HHO	CHE	CHO	DHE	DHO	HHE	HHO
RiskAmount Pr & As = 10%	C7	16.0	14.9	22.6	11.5	23.1	15.4	6.71	6.76	9.99	5.86	10.4	8.41
RiskAmount Pr & As = 15%	C8	14.7	12.5	13.9	7.8	15.1	10.5	6.71	6.93	7.19	4.14	8.47	5.99
RiskAmount_Assemble = 15%	C9	13.4	13.6	16.5	9.1	15.2	10.7	6.63	6.58	8.95	4.96	9.79	5.74
Quality_incr_Chance = 0.01%	C10	20.1	18.9	129.8	39.0	163.7	65.1	1.91	1.93	9.12	3.34	7.72	4.62
Quality_decr_Chance = 25%	C11	20.5	19.8	137.2	40.9	170.0	68.8	2.19	1.88	7.57	2.42	6.50	3.36
Quality_decr_Chance = 75%	C12	19.2	18.8	116.8	37.4	154.7	60.5	2.16	1.75	16.0	3.89	10.3	8.66
Quality_decr_Upper = 0.5	C13	19.1	18.0	96.0	32.3	126.8	53.1	2.28	2.17	16.0	5.33	21.2	7.68
Qual_incr Chance & Upper * 2	C14	20.4	18.7	130.9	39.7	165.8	66.5	2.06	2.67	11.2	2.77	5.85	7.65
RiskQuality_Modifier = 10.0	C15	19.4	18.2	122.3	37.9	159.5	63.2	1.85	2.16	11.8	3.32	11.0	4.83
RiskQuality_Modifier = 25.0	C16	16.9	16.2	103.3	34.5	144.7	56.7	2.42	2.12	15.6	3.97	16.8	5.24
RiskFactory_Modifier = 0.5	C17	19.5	18.7	128.1	39.4	161.9	65.1	3.29	2.38	12.1	2.92	18.1	5.21
RiskFactory_Modifier = 1.0	C18	18.8	18.4	126.9	38.8	163.3	64.5	4.18	2.65	12.4	4.73	17.8	7.91
Quality Thres & Chance	C19	12.1	11.4	27.4	12.4	30.8	18.6	2.04	2.48	7.83	4.09	11.1	6.48
Initial_Printable / 2.0	D1	20.0	18.5	103.0	28.3	138.3	56.4	2.53	2.48	8.79	2.21	11.5	5.25
Initial_Printable * 2.0	D2	19.8	18.5	138.6	57.0	164.6	91.6	2.02	2.05	5.70	4.41	6.09	6.55
Initial_Materials = 0	D3	20.0	18.8	110.8	32.8	156.8	66.3	1.95	2.39	11.9	3.26	12.1	4.41
Initial_Materials / 2.0	D4	20.2	19.4	121.5	36.9	163.9	67.0	2.69	2.16	15.7	3.03	8.69	3.75
Initial_Materials * 2.0	D5	20.4	18.7	136.7	43.3	164.6	65.7	2.12	2.53	10.8	5.46	5.76	7.62
Env_Materials / 2.0	D6	22.5	21.2	95.8	39.5	112.5	69.1	1.76	2.59	3.9	2.91	3.88	4.26
Env_Materials * 2.0	D7	20.1	18.8	125.8	40.0	157.4	65.4	2.22	1.97	11.3	2.95	7.25	7.62
Env_Materials * 100	D8	20.0	19.0	127.9	39.3	156.9	64.9	2.59	1.82	11.3	2.84	6.90	4.82
Initial_NonPr / 2.0	D9	20.0	18.9	66.5	39.4	79.2	64.1	2.01	2.14	3.97	2.92	4.15	3.93
Initial_NonPr * 2.0	D10	20.0	18.9	131.1	39.4	177.9	64.9	2.41	2.09	10.5	4.89	8.59	4.95
Initial NonPr & Env * 2.0	D11	20.0	18.2	127.1	39.5	177.0	66.0	1.83	2.11	10.7	3.01	13.6	4.08
Initial NonPr & Env * 2.0, Raw=0	D12	20.3	18.9	111.9	33.5	154.6	65.8	1.99	2.20	10.3	3.14	14.3	4.16

A.3. Base Algorithm: Time-Step 70

In this subsection, the assembly, print, and collection potentials are provided for time-step 70. Furthermore, the number of robots destroyed, and number of capabilities lost are provided for time-step 70 as well. Table A.7 lists the data for assembly potential. Table A.8 lists the data for print potential. Table A.9 lists the data for collection potential. Table A.10 lists the data for the number of robots destroyed. Table A.11 lists the data for the number of capabilities lost.

Table A.7. Assembly potential and standard deviation for time-step 70.

Base Algorithm	ID	Assembly Potential						Assembly Potential Std Dev					
Time-Step: 70		CHE	CHO	DHE	DHO	HHE	HHO	CHE	CHO	DHE	DHO	HHE	HHO
(Default)	A0	1.00	1.00	73.82	82.95	63.79	70.91	0.00	0.00	4.67	8.64	3.10	3.04
BaseCost_Pr + 1	A1	0.99	0.98	66.70	64.27	55.13	58.06	0.10	0.14	3.55	7.36	2.70	3.86
BaseCost_Pr + 3	A2	1.00	0.99	49.80	43.40	43.91	36.96	0.00	0.10	3.26	5.04	9.75	2.59
BaseCost_Pr + 5	A3	1.00	1.00	32.91	31.08	22.60	32.25	0.00	0.00	4.26	4.02	3.21	4.02
PrintCost_Pr + 1	A4	1.00	0.98	74.39	64.88	63.08	64.86	0.00	0.14	8.61	7.62	2.84	5.54
PrintCost_Pr + 3	A5	0.99	0.98	68.39	43.22	57.32	12.28	0.10	0.14	4.33	4.91	2.89	3.47
PrintCost_Pr + 5	A6	1.00	0.99	81.63	31.83	87.07	8.60	0.00	0.10	15.0	4.22	15.5	0.71
AssembleCost_Pr + 1	A7	0.99	0.99	70.58	65.45	61.10	64.63	0.10	0.10	3.45	6.64	2.63	5.95
AssembleCost_Pr + 3	A8	0.98	1.00	53.52	44.33	50.21	12.08	0.14	0.00	3.49	4.43	2.62	3.52
AssembleCost_Pr + 5	A9	1.00	0.99	44.90	31.69	42.78	8.68	0.00	0.10	3.28	3.53	2.32	0.66
BaseCost_Time + 2	A10	1.00	1.00	75.65	49.08	63.69	36.37	0.00	0.00	5.59	4.70	4.53	2.84
BaseCost_Time - 1	A11	1.00	0.99	74.13	89.25	62.92	72.00	0.00	0.10	3.74	4.16	2.85	3.11
PrintCost_Time + 2	A12	0.99	0.99	75.44	49.41	64.01	45.90	0.10	0.10	4.00	4.82	3.14	6.58
PrintCost_Time - 1	A13	1.00	0.99	75.47	89.33	64.79	71.55	0.00	0.10	4.34	4.86	2.95	2.19
AssembleCost_Time + 2	A14	0.99	1.00	76.41	48.64	65.01	46.38	0.10	0.00	4.29	4.62	2.71	3.84
AssembleCost_Time - 1	A15	0.99	1.00	74.88	89.28	63.67	71.43	0.10	0.00	4.11	4.19	2.32	2.09
Print & Assemble Time + 2	A16	0.99	0.99	74.85	34.03	64.43	28.18	0.10	0.10	5.03	2.86	2.76	1.65
BaseCost_NonPr + 1	A17	0.99	0.98	48.81	64.86	40.24	47.98	0.10	0.14	3.99	3.64	2.47	1.84
PrintCost_NonPr + 1	A18	0.99	0.99	60.56	64.52	53.98	57.35	0.10	0.10	3.48	4.05	2.66	2.06
AssembleCost_NonPr + 1	A19	1.00	1.00	57.41	63.98	52.23	57.56	0.00	0.00	3.56	4.67	2.22	2.07
[All]CostPrintable + 1	A20	1.00	1.00	56.03	42.81	50.57	33.93	0.00	0.00	3.35	5.36	2.59	7.07
[All]CostPrintable + 2	A21	1.00	0.99	42.95	26.45	34.87	7.67	0.00	0.10	3.79	3.70	5.92	0.57
Print & Assemble Pr + 2	A22	0.98	0.99	55.86	36.57	53.66	9.61	0.14	0.10	3.01	4.53	2.46	0.60
Base & Print Pr + 2	A23	1.00	1.00	65.44	36.98	51.09	9.85	0.00	0.00	6.54	4.48	2.62	4.10
Base & Assemble Pr + 2	A24	1.00	1.00	49.31	37.12	46.99	9.36	0.00	0.00	3.09	4.40	3.42	2.44
[All]CostPrintable - 1	A25	1.00	1.00	70.50	85.76	62.77	72.01	0.00	0.00	4.06	5.26	2.47	2.59
AssembleCost_Pr - 1	A26	1.00	1.00	75.03	87.30	63.57	73.69	0.00	0.00	4.28	5.22	3.00	3.46
PrintCost_Pr - 1	A27	0.98	1.00	71.78	88.86	62.42	73.67	0.14	0.00	3.97	4.94	2.67	3.01
BaseCost_Pr - 1	A28	0.99	1.00	72.72	87.92	62.69	66.50	0.10	0.00	3.53	5.29	2.51	3.14
Print_Efficiency = 0.25	B1	0.99	1.00	38.72	25.27	24.18	14.46	0.10	0.00	4.26	3.75	3.35	0.87
Print_Efficiency = 0.5	B2	1.00	0.99	46.72	41.82	47.03	43.17	0.00	0.10	3.29	4.21	5.04	2.54

(continues)

Table A.7. Assembly potential and standard deviation for time-step 70 (continued).

Base Algorithm	ID	Assembly Potential						Assembly Potential Std Dev					
Time-Step: 70		CHE	CHO	DHE	DHO	HHE	HHO	CHE	CHO	DHE	DHO	HHE	HHO
Print_Efficiency = 1.5	B3	0.98	1.00	72.79	87.60	62.45	71.00	0.14	0.00	3.96	5.25	2.94	2.38
Collect_Amount = 0.25	B4	0.99	0.99	59.86	36.52	56.97	42.05	0.10	0.10	6.92	4.70	4.77	4.11
Collect_Amount = 0.5	B5	0.99	1.00	72.88	55.79	62.25	68.92	0.10	0.00	5.08	6.42	3.60	7.67
Collect_Amount = 1.5	B6	0.99	0.98	75.30	90.94	65.09	70.50	0.10	0.14	4.18	4.40	2.63	3.51
Print_Amount = 0.25	B7	0.99	0.98	41.21	40.88	23.49	14.48	0.10	0.14	5.46	4.97	3.72	0.75
Print_Amount = 0.5	B8	1.00	1.00	72.73	58.39	61.07	43.95	0.00	0.00	4.45	5.65	5.67	3.27
Print_Amount = 1.5	B9	0.99	1.00	75.08	88.56	63.62	70.61	0.10	0.00	3.69	5.60	2.78	3.15
Collect & Print Amount = 0.5	B10	1.00	1.00	68.91	45.00	55.97	43.76	0.00	0.00	5.81	4.60	6.10	3.23
QualityThreshold + 0.1	C1	1.00	0.99	72.96	80.18	62.67	69.15	0.00	0.10	4.43	10.2	3.55	3.91
QualityThreshold + 0.2	C2	1.00	0.98	67.73	69.46	59.13	63.62	0.00	0.14	5.81	13.6	4.28	8.14
QualityThreshold + 0.3	C3	1.00	1.00	60.99	51.65	51.86	49.26	0.00	0.00	5.33	11.4	5.41	9.86
QualityThreshold + 0.4	C4	0.98	0.99	50.44	34.03	44.57	29.60	0.14	0.10	7.85	9.12	4.28	10.00
RiskAmount_Print = 1%	C5	1.00	0.99	74.97	81.29	64.50	70.31	0.00	0.10	3.46	9.20	3.19	4.19
RiskAmount_Assemble = 1%	C6	0.98	0.93	65.19	64.16	54.20	58.08	0.14	0.26	5.03	13.9	4.78	8.35
RiskAmount Pr & As = 10%	C7	0.49	0.51	7.40	4.54	3.08	2.30	0.50	0.50	6.66	4.60	3.07	2.22
RiskAmount Pr & As = 15%	C8	0.38	0.35	2.26	2.08	1.64	1.01	0.49	0.48	3.31	2.99	1.89	1.32
RiskAmount_Assemble = 15%	C9	0.39	0.37	1.05	0.70	0.97	0.94	0.49	0.49	1.56	1.20	1.65	1.22
Quality_incr_Chance= 0.01%	C10	0.99	0.99	74.71	81.51	63.01	70.41	0.10	0.10	4.03	9.28	3.13	2.30
Quality_decr_Chance = 25%	C11	0.98	0.98	77.54	88.71	65.77	71.61	0.14	0.14	2.85	6.93	1.78	1.73
Quality_decr_Chance = 75%	C12	1.00	0.99	69.37	74.61	61.01	67.97	0.00	0.10	5.47	11.7	3.84	4.89
Quality_decr_Upper = 0.5	C13	0.99	1.00	63.25	60.55	56.13	57.86	0.10	0.00	6.82	13.4	5.33	10.3
Qual_incr Chance & Upper * 2	C14	0.98	1.00	75.38	83.46	64.06	70.86	0.14	0.00	3.43	9.60	2.57	2.72
RiskQuality_Modifier = 10.0	C15	0.99	1.00	71.14	77.33	63.06	69.93	0.10	0.00	4.76	9.67	2.98	3.27
RiskQuality_Modifier = 25.0	C16	0.98	0.99	65.03	57.82	58.60	68.22	0.14	0.10	6.24	12.0	4.63	4.06
RiskFactory_Modifier = 0.5	C17	0.97	0.98	74.70	83.20	63.73	70.42	0.17	0.14	4.04	8.91	2.82	3.05
RiskFactory_Modifier = 1.0	C18	0.95	0.90	74.34	83.47	64.15	70.88	0.22	0.30	3.87	8.84	2.69	2.62
Quality Thres & Chance	C19	0.99	1.00	29.51	19.64	25.09	15.49	0.10	0.00	9.07	6.98	7.65	8.34
Initial_Printable / 2.0	D1	1.00	1.00	75.87	59.38	64.37	10.81	0.00	0.00	8.36	6.70	2.90	9.85
Initial_Printable * 2.0	D2	0.98	0.99	71.93	88.86	62.85	71.30	0.14	0.10	4.53	4.38	2.50	2.18
Initial_Materials = 0	D3	0.98	1.00	76.38	70.16	64.35	70.62	0.14	0.00	3.91	9.16	2.83	2.54
Initial_Materials / 2.0	D4	0.99	1.00	74.91	77.87	63.54	70.91	0.10	0.00	3.80	7.92	2.92	3.28
Initial_Materials * 2.0	D5	1.00	1.00	73.79	87.73	64.41	70.50	0.00	0.00	4.18	6.19	2.41	2.82
Env_Materials / 2.0	D6	1.00	0.99	52.30	59.44	44.80	48.24	0.00	0.10	3.13	6.78	2.07	1.42
Env_Materials * 2.0	D7	1.00	0.99	65.55	79.21	59.37	69.86	0.00	0.10	5.87	8.60	3.16	3.64
Env_Materials * 100	D8	1.00	1.00	52.71	80.65	43.30	70.69	0.00	0.00	6.21	7.94	4.23	2.69
Initial_NonPr / 2.0	D9	1.00	0.99	35.24	40.38	32.25	36.19	0.00	0.10	3.23	4.07	2.19	1.66
Initial_NonPr * 2.0	D10	1.00	1.00	80.10	83.57	67.99	75.79	0.00	0.00	4.19	8.64	3.04	3.04
Initial NonPr & Env * 2.0	D11	1.00	1.00	136.2	79.71	113.6	80.60	0.00	0.00	6.62	11.8	12.6	5.10
Initial NonPr & Env * 2.0, Raw=0	D12	1.00	0.99	132.3	70.20	111.3	80.58	0.00	0.10	5.91	7.91	4.27	5.09

Table A.8. Print potential and standard deviation for time-step 70.

Base Algorithm	ID	Print Potential						Print Potential Std Dev					
Time-Step: 70		CHE	CHO	DHE	DHO	HHE	HHO	CHE	CHO	DHE	DHO	HHE	HHO
(Default)	A0	1.00	1.00	62.10	82.99	53.81	70.67	0.00	0.00	3.64	8.47	2.08	2.89
BaseCost_Pr + 1	A1	1.00	1.00	52.91	64.30	44.05	57.84	0.00	0.00	2.41	7.17	1.48	3.77
BaseCost_Pr + 3	A2	1.00	1.00	30.35	43.23	6.44	36.54	0.00	0.00	9.13	4.87	0.70	2.78
BaseCost_Pr + 5	A3	0.99	1.00	5.56	30.97	4.74	31.88	0.10	0.00	0.69	3.93	0.48	4.00
PrintCost_Pr + 1	A4	0.99	1.00	58.28	64.81	52.29	64.59	0.10	0.00	6.56	7.28	2.21	5.50
PrintCost_Pr + 3	A5	0.99	1.00	46.18	43.09	42.55	12.21	0.10	0.00	2.32	4.82	1.42	3.66
PrintCost_Pr + 5	A6	0.99	1.00	27.88	31.49	10.42	8.62	0.10	0.00	11.2	3.99	3.81	0.55
AssembleCost_Pr + 1	A7	1.00	1.00	61.38	65.50	53.65	64.50	0.00	0.00	2.88	6.44	2.41	5.89
AssembleCost_Pr + 3	A8	1.00	1.00	53.50	44.26	45.33	12.15	0.00	0.00	3.49	4.15	4.07	3.81
AssembleCost_Pr + 5	A9	1.00	1.00	44.07	31.48	38.49	8.69	0.00	0.00	3.39	3.52	9.09	0.61
BaseCost_Time + 2	A10	1.00	1.00	60.06	49.12	53.25	36.61	0.00	0.00	4.33	4.38	3.75	2.64
BaseCost_Time - 1	A11	1.00	1.00	64.56	89.06	53.84	71.35	0.00	0.00	2.84	4.01	1.93	3.24
PrintCost_Time + 2	A12	0.99	1.00	61.15	49.55	52.18	46.23	0.10	0.00	2.76	4.62	2.40	6.40
PrintCost_Time - 1	A13	0.99	1.00	61.87	88.98	53.33	71.30	0.10	0.00	3.46	4.87	2.67	2.04
AssembleCost_Time + 2	A14	0.99	1.00	59.10	48.79	53.98	46.72	0.10	0.00	3.43	4.28	2.62	3.60
AssembleCost_Time - 1	A15	0.97	1.00	63.26	89.30	54.15	71.31	0.17	0.00	3.05	4.00	2.03	2.00
Print & Assemble Time + 2	A16	0.99	1.00	60.01	34.04	55.00	28.33	0.10	0.00	3.54	2.81	2.34	1.36
BaseCost_NonPr + 1	A17	1.00	1.00	41.33	64.59	33.68	47.77	0.00	0.00	2.25	3.47	1.41	1.52
PrintCost_NonPr + 1	A18	0.98	1.00	50.47	64.28	45.54	57.08	0.14	0.00	2.71	3.95	1.68	1.99
AssembleCost_NonPr + 1	A19	1.00	1.00	48.54	63.92	43.72	57.09	0.00	0.00	2.53	4.49	2.06	2.17
[All]CostPrintable + 1	A20	0.98	1.00	42.53	42.64	31.39	33.55	0.14	0.00	1.75	5.06	8.66	7.10
[All]CostPrintable + 2	A21	0.99	1.00	6.57	26.26	6.43	7.57	0.10	0.00	0.59	3.57	0.74	0.59
Print & Assemble Pr + 2	A22	0.98	1.00	42.45	36.55	11.49	9.62	0.14	0.00	2.17	4.40	7.84	0.75
Base & Print Pr + 2	A23	1.00	1.00	11.10	36.81	7.48	9.80	0.00	0.00	7.43	4.32	0.67	4.01
Base & Assemble Pr + 2	A24	0.98	1.00	26.92	36.89	11.11	9.22	0.14	0.00	9.79	4.36	6.07	2.32
[All]CostPrintable - 1	A25	0.99	1.00	65.45	85.88	55.97	71.96	0.10	0.00	3.34	4.85	1.93	2.61
AssembleCost_Pr - 1	A26	1.00	1.00	62.13	87.18	54.69	73.51	0.00	0.00	3.40	5.14	2.30	3.48
PrintCost_Pr - 1	A27	0.99	1.00	63.75	88.59	55.97	73.49	0.10	0.00	4.11	4.84	2.37	2.96
BaseCost_Pr - 1	A28	1.00	1.00	63.51	87.65	55.77	66.49	0.00	0.00	3.45	5.21	2.09	3.18
Print_Efficiency = 0.25	B1	0.98	1.00	11.48	24.91	10.23	14.38	0.14	0.00	1.18	3.59	0.85	0.74
Print_Efficiency = 0.5	B2	1.00	1.00	37.23	41.55	24.31	42.75	0.00	0.00	2.00	4.14	8.54	2.67
Print_Efficiency = 1.5	B3	1.00	1.00	63.59	87.46	55.53	70.86	0.00	0.00	3.67	4.82	2.10	2.37
Collect_Amount = 0.25	B4	0.99	1.00	42.11	36.65	44.21	42.20	0.10	0.00	6.10	4.53	3.89	3.76
Collect_Amount = 0.5	B5	0.98	1.00	57.16	55.65	49.88	68.98	0.14	0.00	3.83	6.21	2.35	7.58
Collect_Amount = 1.5	B6	0.99	1.00	63.92	90.73	54.23	70.42	0.10	0.00	3.38	4.33	1.96	3.07
Print_Amount = 0.25	B7	0.99	1.00	11.83	40.04	10.06	14.13	0.10	0.00	1.43	4.84	0.96	0.95
Print_Amount = 0.5	B8	0.99	1.00	63.94	57.85	33.05	43.45	0.10	0.00	3.56	5.35	13.2	3.39
Print_Amount = 1.5	B9	1.00	1.00	62.82	88.72	53.13	70.87	0.00	0.00	3.56	5.61	2.19	3.06
Collect & Print Amount = 0.5	B10	0.99	1.00	59.81	44.67	32.86	43.58	0.10	0.00	3.61	4.60	13.0	3.21
QualityThreshold + 0.1	C1	0.99	1.00	60.75	80.18	52.18	68.92	0.10	0.00	3.69	10.1	2.73	3.83
QualityThreshold + 0.2	C2	1.00	1.00	55.24	69.43	48.35	63.48	0.00	0.00	5.06	13.3	3.85	8.07
QualityThreshold + 0.3	C3	0.97	1.00	48.96	51.72	42.20	49.38	0.17	0.00	4.88	11.3	6.97	9.71
QualityThreshold + 0.4	C4	1.00	1.00	39.59	34.00	35.26	29.65	0.00	0.00	6.53	9.17	4.69	9.97
RiskAmount_Print = 1%	C5	0.93	1.00	55.53	73.98	45.09	62.55	0.26	0.00	4.48	10.0	3.48	5.22
RiskAmount_Assemble = 1%	C6	0.99	1.00	59.30	70.22	51.31	65.23	0.10	0.00	3.93	12.8	2.46	7.61

(continues)

Table A.8. Print potential and standard deviation for time-step 70 (continued).

Base Algorithm	ID	Print Potential						Print Potential Std Dev					
Time-Step: 70		CHE	CHO	DHE	DHO	HHE	HHO	CHE	CHO	DHE	DHO	HHE	HHO
RiskAmount Pr & As = 10%	C7	0.49	0.83	0.92	2.95	0.77	2.59	0.50	0.38	0.99	2.58	0.89	2.19
RiskAmount Pr & As = 15%	C8	0.36	0.73	0.36	1.37	0.45	1.42	0.48	0.45	0.50	1.44	0.70	1.39
RiskAmount_Assemble = 15%	C9	1.00	1.00	6.60	9.14	5.66	8.65	0.00	0.00	4.11	6.63	3.81	5.92
Quality_incr_Chance = 0.01%	C10	1.00	1.00	62.43	81.45	53.54	70.38	0.00	0.00	3.13	9.18	1.96	2.24
Quality_decr_Chance = 25%	C11	0.97	1.00	64.80	88.80	54.73	71.69	0.17	0.00	2.21	6.85	1.34	1.35
Quality_decr_Chance = 75%	C12	0.99	1.00	58.40	74.43	50.94	67.81	0.10	0.00	4.86	11.5	3.57	4.88
Quality_decr_Upper = 0.5	C13	0.99	1.00	52.63	60.63	45.77	57.99	0.10	0.00	5.49	13.1	5.86	10.0
Qual_incr Chance & Upper * 2	C14	0.99	1.00	63.34	83.49	53.84	70.84	0.10	0.00	3.12	9.35	2.00	2.36
RiskQuality_Modifier = 10.0	C15	0.99	1.00	60.25	77.37	53.42	69.83	0.10	0.00	4.04	9.42	2.13	3.08
RiskQuality_Modifier = 25.0	C16	1.00	1.00	56.36	58.15	50.87	68.15	0.00	0.00	5.31	11.6	3.15	3.71
RiskFactory_Modifier = 0.5	C17	0.97	1.00	62.34	83.07	53.20	70.46	0.17	0.00	3.26	8.90	1.83	2.76
RiskFactory_Modifier = 1.0	C18	0.94	0.99	62.48	83.47	53.50	70.97	0.24	0.10	2.90	8.64	2.29	2.29
Quality Thres & Chance	C19	1.00	1.00	20.55	19.60	17.87	15.58	0.00	0.00	8.18	6.89	8.65	8.29
Initial_Printable / 2.0	D1	0.99	1.00	61.19	59.33	52.70	10.82	0.10	0.00	6.82	6.72	1.71	9.86
Initial_Printable * 2.0	D2	0.99	1.00	63.81	88.78	54.94	71.11	0.10	0.00	3.53	4.15	2.61	2.21
Initial_Materials = 0	D3	1.00	1.00	61.34	70.10	52.42	70.50	0.00	0.00	3.02	8.91	2.09	2.32
Initial_Materials / 2.0	D4	1.00	1.00	62.96	78.02	53.38	70.74	0.00	0.00	3.68	7.80	2.23	3.35
Initial_Materials * 2.0	D5	1.00	1.00	63.03	87.63	54.15	70.26	0.00	0.00	3.16	5.85	1.77	2.47
Env_Materials / 2.0	D6	1.00	1.00	42.24	59.77	36.04	48.43	0.00	0.00	1.79	6.63	1.30	1.18
Env_Materials * 2.0	D7	1.00	1.00	60.73	79.24	52.92	69.42	0.00	0.00	3.47	8.23	2.44	3.60
Env_Materials * 100	D8	0.99	1.00	59.80	80.74	51.50	69.99	0.10	0.00	3.60	7.83	2.26	2.56
Initial_NonPr / 2.0	D9	1.00	1.00	31.02	40.10	26.46	35.96	0.00	0.00	2.11	3.94	1.47	1.53
Initial_NonPr * 2.0	D10	0.98	1.00	68.28	83.61	58.46	75.69	0.14	0.00	3.29	8.49	2.19	2.95
Initial NonPr & Env * 2.0	D11	1.00	1.00	121.4	79.74	102.6	80.22	0.00	0.00	5.81	11.4	11.0	4.88
Initial NonPr & Env * 2.0, Raw=0	D12	0.97	0.99	116.9	70.08	98.49	80.26	0.17	0.10	4.72	7.58	3.65	4.95

Table A.9. Collection potential and standard deviation for time-step 70.

Base Algorithm	ID	Collection Potential						Collection Potential Std Dev					
Time-Step: 70		CHE	CHO	DHE	DHO	HHE	HHO	CHE	CHO	DHE	DHO	HHE	HHO
(Default)	A0	29.2	28.2	137.4	83.9	165.5	142.5	2.85	2.19	6.86	8.50	6.04	6.29
BaseCost_Pr + 1	A1	29.3	26.9	120.9	64.9	136.8	118.4	2.77	3.38	4.89	7.09	4.86	6.75
BaseCost_Pr + 3	A2	28.2	17.9	81.2	43.7	60.9	70.7	2.75	3.19	9.35	4.90	3.65	5.65
BaseCost_Pr + 5	A3	18.8	12.8	39.0	31.4	37.6	47.0	2.53	2.41	4.35	3.92	3.71	5.95
PrintCost_Pr + 1	A4	29.5	28.2	134.1	65.6	166.3	142.6	2.56	3.10	14.4	7.17	6.71	7.49
PrintCost_Pr + 3	A5	29.4	28.4	115.7	43.6	145.4	129.9	2.43	3.90	4.89	4.77	5.65	7.02
PrintCost_Pr + 5	A6	29.5	28.1	110.9	32.0	159.2	105.1	2.84	3.33	7.52	4.07	24.7	6.57
AssembleCost_Pr + 1	A7	29.2	28.1	133.4	66.2	161.6	143.0	3.32	3.33	4.40	6.36	5.98	6.62
AssembleCost_Pr + 3	A8	29.2	28.3	108.3	44.6	135.0	130.9	3.94	2.31	5.08	4.29	6.98	7.76
AssembleCost_Pr + 5	A9	28.7	28.2	89.9	31.9	115.3	105.3	2.86	3.60	3.64	3.38	7.68	7.86
BaseCost_Time + 2	A10	13.4	12.6	137.5	49.7	161.3	74.6	1.71	1.79	8.61	4.36	9.53	5.91
BaseCost_Time - 1	A11	64.0	50.0	140.0	89.9	166.8	140.2	2.63	3.05	5.38	3.87	7.33	7.86
PrintCost_Time + 2	A12	29.3	28.5	138.1	50.1	165.2	95.6	3.48	3.21	5.43	4.52	5.85	12.4
PrintCost_Time - 1	A13	29.2	28.2	138.9	89.9	163.0	143.6	2.80	3.41	6.25	4.68	5.69	6.57
AssembleCost_Time + 2	A14	29.5	28.0	137.4	49.3	163.1	97.9	3.24	2.73	6.21	4.38	5.31	6.16
AssembleCost_Time - 1	A15	29.0	28.2	139.6	90.1	167.4	143.9	3.76	2.70	5.53	3.96	5.79	6.72
Print & Assemble Time + 2	A16	29.2	28.0	136.7	34.4	162.5	65.9	3.28	2.62	7.47	2.65	6.65	7.01
BaseCost_NonPr + 1	A17	29.0	28.7	91.3	65.4	102.3	90.7	3.14	2.74	4.54	3.33	4.28	4.57
PrintCost_NonPr + 1	A18	29.0	28.1	112.5	65.1	139.7	114.0	3.69	2.81	4.86	3.88	5.94	5.53
AssembleCost_NonPr + 1	A19	29.8	28.1	107.3	64.7	134.4	113.5	2.53	2.66	4.36	4.40	6.25	5.06
[All]CostPrintable + 1	A20	29.3	27.3	99.7	43.3	116.8	90.7	2.36	2.76	4.03	4.93	6.52	5.46
[All]CostPrintable + 2	A21	28.9	22.6	50.0	26.6	58.7	64.4	2.84	2.66	3.81	3.64	3.41	4.93
Print & Assemble Pr + 2	A22	29.3	28.5	99.5	36.9	140.4	113.1	3.28	2.48	3.97	4.43	15.1	7.31
Base & Print Pr + 2	A23	29.5	22.6	77.4	37.3	80.2	71.0	2.24	2.79	7.27	4.36	6.41	9.17
Base & Assemble Pr + 2	A24	28.8	22.0	77.2	37.4	77.1	72.1	2.81	2.68	10.3	4.27	6.95	5.01
[All]CostPrintable - 1	A25	28.4	28.4	137.8	86.7	163.9	138.1	2.74	2.53	5.70	4.88	6.17	5.29
AssembleCost_Pr - 1	A26	29.4	28.2	138.5	88.2	164.1	137.7	2.44	2.50	5.70	4.89	6.21	5.99
PrintCost_Pr - 1	A27	28.9	28.5	137.3	89.6	163.1	138.9	3.69	2.55	6.68	4.65	6.66	6.17
BaseCost_Pr - 1	A28	29.1	28.2	137.6	88.7	164.2	151.2	2.73	2.80	5.84	5.04	6.94	6.49
Print_Efficiency = 0.25	B1	29.5	28.3	51.2	25.4	68.9	72.9	3.13	2.39	4.86	3.66	3.45	5.00
Print_Efficiency = 0.5	B2	29.2	28.8	85.0	42.1	101.3	90.0	2.54	2.45	3.71	4.13	5.44	4.68
Print_Efficiency = 1.5	B3	29.3	28.2	137.8	88.4	163.0	143.2	3.03	2.73	6.30	4.87	7.18	5.08
Collect_Amount = 0.25	B4	25.1	24.1	102.9	36.9	139.6	82.6	2.35	2.43	12.0	4.52	10.7	12.6
Collect_Amount = 0.5	B5	24.8	24.2	131.6	56.2	157.4	132.7	2.90	2.80	7.45	6.27	7.59	15.5
Collect_Amount = 1.5	B6	31.7	30.3	140.7	91.7	169.5	146.1	3.08	3.35	6.13	4.05	4.8	5.93
Print_Amount = 0.25	B7	29.6	28.2	54.0	41.1	69.0	72.2	2.56	3.45	6.47	4.81	3.64	4.78
Print_Amount = 0.5	B8	29.3	27.7	138.8	59.0	138.1	96.2	2.62	2.87	5.78	5.34	22.8	7.52
Print_Amount = 1.5	B9	29.2	28.4	139.2	89.3	163.9	142.3	3.32	2.34	5.79	5.41	6.32	7.12
Collect & Print Amount = 0.5	B10	24.3	24.2	130.7	45.3	128.7	88.5	2.56	2.47	7.85	4.45	22.3	6.76
QualityThreshold + 0.1	C1	29.0	28.6	135.1	81.0	161.4	138.1	2.63	2.37	6.82	10.1	8.29	7.44
QualityThreshold + 0.2	C2	29.4	27.5	124.3	70.1	151.2	126.0	2.55	4.58	9.29	13.4	10.5	14.9
QualityThreshold + 0.3	C3	26.2	25.6	111.0	52.2	131.1	101.9	3.11	2.76	8.24	11.4	15.4	17.9
QualityThreshold + 0.4	C4	20.4	19.1	90.8	34.3	111.8	64.0	4.01	3.09	13.1	9.22	9.01	18.4
RiskAmount_Print = 1%	C5	29.5	28.2	139.0	82.1	166.8	142.3	2.52	3.52	5.18	9.07	6.02	7.10
RiskAmount_Assemble = 1%	C6	29.0	26.8	132.2	71.0	159.2	134.2	3.68	5.84	6.31	12.8	6.89	15.3

(continues)

Table A.9. Collection potential and standard deviation for time-step 70 (continued).

Base Algorithm	ID	Collection Potential						Collection Potential Std Dev					
Time-Step: 70		CHE	CHO	DHE	DHO	HHE	HHO	CHE	CHO	DHE	DHO	HHE	HHO
RiskAmount Pr & As = 10%	C7	20.2	19.6	27.7	12.4	27.1	21.9	11.0	11.1	12.9	7.09	15.7	13.7
RiskAmount Pr & As = 15%	C8	18.3	15.0	14.5	7.8	16.3	12.3	10.8	11.3	9.28	4.59	9.60	8.39
RiskAmount_Assemble = 15%	C9	17.9	16.5	17.2	9.3	16.9	12.3	11.0	10.8	10.5	6.68	12.5	9.09
Quality_incr_Chance = 0.01%	C10	29.0	28.6	138.7	82.3	165.0	141.5	3.77	2.66	5.70	9.15	6.26	7.24
Quality_decr_Chance = 25%	C11	30.1	28.4	143.5	89.4	170.7	148.1	2.82	3.87	3.15	6.71	3.89	3.68
Quality_decr_Chance = 75%	C12	28.7	27.2	129.5	75.3	156.0	133.8	2.65	3.48	8.48	11.5	8.99	9.89
Quality_decr_Upper = 0.5	C13	28.4	27.1	117.5	61.4	142.8	115.4	2.72	2.89	10.9	13.2	12.2	17.2
Qual_incr Chance & Upper * 2	C14	28.7	28.0	139.9	84.3	166.8	143.6	2.89	2.77	5.16	9.26	5.61	6.56
RiskQuality_Modifier = 10.0	C15	27.9	27.5	133.3	78.4	162.7	138.1	3.48	2.69	7.50	9.28	8.05	7.39
RiskQuality_Modifier = 25.0	C16	25.2	23.3	123.8	59.3	151.2	125.7	4.38	4.01	10.3	11.6	11.5	9.99
RiskFactory_Modifier = 0.5	C17	28.7	27.5	138.8	84.0	165.4	142.1	3.09	3.75	5.62	8.74	5.58	6.56
RiskFactory_Modifier = 1.0	C18	28.0	26.5	138.4	84.2	165.7	143.7	5.05	7.48	4.78	8.59	6.16	5.66
Quality Thres & Chance	C19	15.7	14.5	50.8	19.8	58.5	33.2	3.43	3.51	16.5	6.97	21.0	12.4
Initial_Printable / 2.0	D1	28.9	23.1	138.6	59.9	166.2	96.4	2.64	2.50	14.7	6.53	6.03	6.89
Initial_Printable * 2.0	D2	29.0	28.3	137.2	89.5	164.0	140.8	3.59	3.13	6.38	4.01	6.24	6.65
Initial_Materials = 0	D3	29.2	28.2	139.0	70.8	166.7	143.0	4.08	2.63	5.22	8.98	5.90	5.43
Initial_Materials / 2.0	D4	29.0	28.5	139.4	78.6	165.0	143.3	3.48	2.48	6.08	7.73	6.22	6.31
Initial_Materials * 2.0	D5	29.0	28.6	138.3	88.6	166.4	141.7	2.79	2.27	6.22	5.90	5.63	6.80
Env_Materials / 2.0	D6	32.7	31.4	95.3	60.1	113.0	97.0	2.13	2.45	3.37	6.62	3.61	3.95
Env_Materials * 2.0	D7	24.6	24.1	128.6	80.1	156.2	133.1	2.11	2.41	7.81	8.27	6.24	8.55
Env_Materials * 100	D8	24.7	24.2	115.4	81.4	127.8	126.7	2.66	2.07	8.09	7.82	7.42	6.90
Initial_NonPr / 2.0	D9	29.4	28.3	67.2	40.7	79.6	67.2	2.13	2.44	3.91	3.91	4.00	4.88
Initial_NonPr * 2.0	D10	29.6	28.0	149.9	84.2	179.3	156.5	2.61	2.91	6.24	8.51	7.74	6.34
Initial NonPr & Env * 2.0	D11	24.3	23.5	260.4	80.7	310.4	174.9	2.63	2.44	10.8	11.6	33.5	13.5
Initial NonPr & Env * 2.0, Raw=0	D12	24.7	23.8	252.0	70.8	301.8	174.6	2.67	3.19	8.66	7.73	10.4	15.3

Table A.10. Number of robots destroyed due to build quality and task hazards.

(70)	Destroyed (Build Quality)						Destroyed (Hazard)					
	CHE	CHO	DHE	DHO	HHE	HHO	CHE	CHO	DHE	DHO	HHE	HHO
C1	0.0	0.0	7.1	4.1	8.8	7.7	7.0	6.4	8.9	8.0	9.0	9.3
C2	0.0	0.0	18.0	8.1	19.5	17.1	6.6	6.9	8.8	7.9	8.3	8.7
C3	3.5	3.6	32.7	14.2	38.2	28.6	6.3	5.9	7.5	5.8	7.5	7.2
C4	9.6	10.3	47.9	18.0	58.6	34.6	5.7	5.5	6.0	3.0	5.8	5.7
C5	0.0	0.0	3.0	1.6	2.9	2.3	6.5	6.5	9.1	9.5	9.2	9.7
C6	0.0	0.0	2.7	1.2	2.6	2.4	6.7	6.7	8.8	8.8	8.9	8.8
C7	0.0	0.0	0.0	0.0	0.0	0.0	5.6	5.5	6.6	4.0	6.2	6.2
C8	0.0	0.0	0.0	0.0	0.0	0.0	5.5	5.0	5.0	2.9	5.4	4.5
C9	0.0	0.0	0.0	0.0	0.0	0.0	5.4	5.5	4.5	1.7	4.9	2.7
C10	0.0	0.0	3.2	1.9	3.9	4.1	6.7	6.3	9.2	9.3	10.2	10.1
C11	0.0	0.0	0.3	0.1	0.6	0.5	5.6	6.2	7.6	7.4	7.7	7.1
C12	0.0	0.0	10.0	4.4	11.3	9.5	7.3	7.6	11.5	10.4	11.5	11.4
C13	0.0	0.0	22.5	10.2	24.7	22.2	7.4	7.9	10.9	8.6	10.8	10.7
C14	0.0	0.0	2.6	1.2	2.7	2.6	7.1	7.0	8.9	9.1	9.3	9.0
C15	0.0	0.0	3.3	1.1	3.2	3.3	8.0	7.5	14.1	12.2	12.9	13.9
C16	0.0	0.0	1.8	0.9	2.8	2.8	10.5	11.4	24.5	20.3	24.2	24.9
C17	0.0	0.0	3.2	1.3	3.6	3.1	7.0	7.1	9.0	9.4	9.9	10.0
C18	0.0	0.0	3.1	1.1	3.3	2.5	7.0	6.0	9.6	9.3	10.0	9.4
C19	15.3	15.2	57.1	20.2	68.2	39.2	5.0	5.3	4.1	1.6	5.5	4.3

Table A.11. Total number of robots and total number of capabilities lost.

(70)	Number of Robots Total						Capabilities Lost					
	CHE	CHO	DHE	DHO	HHE	HHO	CHE	CHO	DHE	DHO	HHE	HHO
C1	36.0	35.0	151.1	93.1	179.2	155.0	0.0	0.0	1.7	1.7	1.6	1.8
C2	36.0	34.4	151.1	86.1	179.0	151.8	0.0	0.0	1.6	1.4	1.6	1.7
C3	36.0	35.0	151.3	72.2	176.8	137.7	0.0	0.0	1.2	1.3	1.3	1.3
C4	35.7	35.0	144.7	55.3	176.3	104.3	0.0	0.0	1.0	0.7	1.1	0.8
C5	36.0	34.8	151.1	93.2	178.8	154.3	0.1	0.0	10.0	10.3	9.9	9.7
C6	35.7	33.5	143.8	81.0	170.7	145.4	0.0	0.1	9.1	7.9	9.5	8.8
C7	25.8	25.2	34.4	16.4	33.4	28.1	1.0	0.7	25.4	24.5	20.9	21.6
C8	23.8	19.9	19.5	10.6	21.7	16.7	1.3	0.9	16.7	17.6	16.4	16.9
C9	23.2	22.0	21.7	10.9	21.8	15.0	0.6	0.6	14.0	10.4	11.9	8.8
C10	35.7	34.9	151.1	93.5	179.0	155.7	0.0	0.0	1.8	1.8	1.8	1.8
C11	35.8	34.6	151.4	96.9	179.0	155.7	0.1	0.0	1.3	1.4	1.3	1.6
C12	36.0	34.8	151.0	90.1	178.8	154.7	0.0	0.0	2.1	1.8	2.2	2.0
C13	35.8	35.0	150.9	80.2	178.3	148.2	0.0	0.0	2.2	1.8	1.9	2.0
C14	35.8	35.0	151.3	94.5	178.8	155.2	0.0	0.0	1.4	1.7	1.7	2.0
C15	35.9	35.0	150.7	91.6	178.8	155.2	0.0	0.0	2.5	2.2	2.0	2.4
C16	35.6	34.7	150.1	80.4	178.2	153.4	0.0	0.0	3.9	3.3	4.1	4.9
C17	35.7	34.5	151.0	94.7	178.9	155.1	0.1	0.0	2.0	1.9	2.0	1.8
C18	35.0	32.6	151.1	94.6	179.0	155.5	0.1	0.1	1.8	1.7	1.6	1.8
C19	35.9	35.0	111.9	41.6	132.2	76.7	0.0	0.0	0.8	0.3	0.9	0.5

A.4. Base Algorithm: Averaged Across Time-Steps

In this subsection, select secondary metrics are listed for certain experimental condition categories. Furthermore, the data in this subsection consists of ratios and percentages, which is averaged across the time-step values of 30, 50, and 70. Table A.12 lists the average print and assemble ratios for experimental condition classification 'A'. Table A.13 lists the average robot quality for experimental condition classification 'C'.

Table A.12. Average of the print ratios and assemble ratios across all time-steps.

Average	**Print Ratio**						**Assemble Ratio**					
	CHE	**CHO**	**DHE**	**DHO**	**HHE**	**HHO**	**CHE**	**CHO**	**DHE**	**DHO**	**HHE**	**HHO**
A0	0.05	0.06	0.45	0.99	0.32	0.53	0.05	0.06	0.54	0.99	0.39	0.53
A1	0.05	0.06	0.43	0.99	0.32	0.50	0.05	0.06	0.56	0.99	0.42	0.50
A2	0.06	0.07	0.36	0.99	0.18	0.50	0.05	0.07	0.63	0.99	0.67	0.51
A3	0.06	0.08	0.22	0.99	0.19	0.67	0.06	0.08	0.77	0.99	0.57	0.67
A4	0.06	0.06	0.43	0.99	0.31	0.50	0.05	0.06	0.56	0.99	0.39	0.50
A5	0.06	0.06	0.40	0.99	0.30	0.25	0.05	0.06	0.59	0.99	0.40	0.25
A6	0.06	0.06	0.27	0.99	0.15	0.19	0.05	0.06	0.72	1.00	0.50	0.19
A7	0.05	0.06	0.45	0.99	0.32	0.50	0.05	0.06	0.54	0.99	0.39	0.50
A8	0.05	0.06	0.48	0.99	0.31	0.25	0.05	0.06	0.50	0.99	0.40	0.25
A9	0.05	0.06	0.44	0.99	0.34	0.19	0.05	0.06	0.55	0.99	0.39	0.19
A10	0.10	0.11	0.42	0.99	0.31	0.60	0.10	0.11	0.57	0.99	0.43	0.59
A11	0.03	0.03	0.46	0.99	0.32	0.49	0.03	0.03	0.53	0.99	0.38	0.49
A12	0.05	0.06	0.40	0.99	0.29	0.56	0.05	0.06	0.59	0.99	0.42	0.55
A13	0.05	0.06	0.44	0.99	0.33	0.51	0.05	0.06	0.55	0.99	0.41	0.51
A14	0.05	0.06	0.42	0.99	0.34	0.55	0.05	0.06	0.56	0.99	0.40	0.54
A15	0.06	0.06	0.43	0.99	0.31	0.51	0.05	0.06	0.55	0.99	0.39	0.51
A16	0.06	0.06	0.42	1.00	0.31	0.53	0.05	0.06	0.57	0.98	0.42	0.52
A17	0.05	0.06	0.45	0.99	0.32	0.54	0.05	0.06	0.54	0.99	0.40	0.54
A18	0.06	0.06	0.45	0.99	0.32	0.53	0.05	0.06	0.54	0.99	0.39	0.53
A19	0.05	0.06	0.45	0.99	0.32	0.53	0.05	0.06	0.54	0.99	0.40	0.53
A20	0.05	0.06	0.41	0.99	0.26	0.39	0.05	0.06	0.58	0.99	0.45	0.39
A21	0.06	0.06	0.22	0.99	0.19	0.25	0.05	0.06	0.77	0.99	0.56	0.24
A22	0.05	0.06	0.41	0.99	0.15	0.21	0.05	0.06	0.58	0.99	0.40	0.20
A23	0.05	0.06	0.22	0.99	0.17	0.26	0.05	0.06	0.77	0.99	0.58	0.26
A24	0.05	0.06	0.33	0.99	0.25	0.26	0.05	0.06	0.66	0.99	0.55	0.26
A25	0.05	0.06	0.47	1.00	0.33	0.51	0.05	0.06	0.52	0.99	0.39	0.50
A26	0.05	0.06	0.45	0.99	0.32	0.55	0.05	0.06	0.54	0.99	0.39	0.55
A27	0.06	0.06	0.46	0.99	0.33	0.55	0.05	0.06	0.53	0.99	0.39	0.54
A28	0.06	0.06	0.46	0.99	0.33	0.46	0.05	0.06	0.53	0.99	0.39	0.45

Table A.13. Average robot quality across all time-steps.

Average	**Average Robot Quality**					
	CHE	**CHO**	**DHE**	**DHO**	**HHE**	**HHO**
C1	0.944	0.942	0.852	0.866	0.852	0.855
C2	0.944	0.942	0.864	0.878	0.862	0.863
C3	0.943	0.941	0.891	0.897	0.890	0.892
C4	0.945	0.942	0.925	0.926	0.926	0.924
C5	0.943	0.942	0.851	0.865	0.849	0.852
C6	0.944	0.943	0.853	0.872	0.857	0.859
C7	0.948	0.944	0.923	0.927	0.920	0.919
C8	0.950	0.945	0.936	0.943	0.937	0.929
C9	0.950	0.946	0.938	0.945	0.939	0.929
C10	0.941	0.940	0.837	0.855	0.836	0.835
C11	0.971	0.971	0.920	0.932	0.921	0.926
C12	0.916	0.912	0.775	0.798	0.771	0.775
C13	0.892	0.885	0.744	0.757	0.745	0.742
C14	0.946	0.944	0.861	0.874	0.858	0.860
C15	0.944	0.942	0.851	0.867	0.849	0.850
C16	0.944	0.939	0.859	0.871	0.858	0.858
C17	0.945	0.940	0.846	0.861	0.845	0.849
C18	0.944	0.941	0.843	0.866	0.844	0.853
C19	0.914	0.910	0.889	0.892	0.889	0.887

APPENDIX B. DECISION-MAKING RESULT TABLES

B.1. Results – Cycle

Table B.1 lists the results for time-step 70. Table B.2 lists the results for time-step 50. Table B.3 lists the results for time-step 30.

Table B.1. Cycle decision-making algorithm results for time-step 70.

Cycle Algorithm	ID	Collection Potential			Assembly Potential			Print Potential		
Time-Step: 70		**DHE**	**HHE**	**HHO**	**DHE**	**HHE**	**HHO**	**DHE**	**HHE**	**HHO**
(Default)	**A0**	137.5	167.5	149.2	68.1	57.6	54.8	67.9	57.0	54.7
BaseCost_Pr + 1	**A1**	119.7	139.2	98.6	58.4	45.9	27.9	60.2	45.9	27.6
BaseCost_Pr + 3	**A2**	82.4	94.2	57.9	39.6	29.9	19.3	41.9	30.5	19.1
BaseCost_Pr + 5	**A3**	63.3	68.2	35.1	29.0	19.3	5.9	33.5	20.6	5.8
PrintCost_Pr + 1	**A4**	133.5	166.7	133.7	66.8	56.3	44.5	65.2	55.3	44.3
PrintCost_Pr + 3	**A5**	111.4	146.4	112.9	58.8	48.8	9.5	51.3	44.1	9.5
PrintCost_Pr + 5	**A6**	97.9	134.6	97.7	55.4	45.4	7.7	41.4	35.6	7.7
AssembleCost_Pr + 1	**A7**	132.0	164.0	134.3	63.2	55.1	45.3	67.5	55.9	45.0
AssembleCost_Pr + 3	**A8**	110.5	141.8	114.4	48.7	44.1	9.6	60.3	50.1	9.7
AssembleCost_Pr + 5	**A9**	98.3	129.5	97.7	39.2	36.6	7.7	57.9	46.2	7.9
BaseCost_Time + 2	**A10**	128.9	101.0	64.2	64.9	35.2	20.4	62.2	34.3	20.7
BaseCost_Time - 1	**A11**	138.6	171.4	155.0	68.3	56.3	66.8	69.1	56.0	66.4
PrintCost_Time + 2	**A12**	133.5	158.8	96.1	65.8	55.0	31.2	65.9	54.4	31.3
PrintCost_Time - 1	**A13**	137.8	163.4	155.5	68.1	56.2	63.6	68.4	55.2	63.7
AssembleCost_Time + 2	**A14**	136.4	164.0	94.8	68.7	56.5	30.8	66.2	56.7	30.9
AssembleCost_Time - 1	**A15**	135.7	167.8	154.9	66.7	55.8	64.3	67.7	56.3	64.0
Print & Assemble Time + 2	**A16**	128.0	154.5	63.8	65.1	52.3	20.6	61.2	51.8	20.8
BaseCost_NonPr + 1	**A17**	89.9	101.8	100.1	43.9	35.7	39.2	44.8	35.9	38.7
PrintCost_NonPr + 1	**A18**	107.9	137.9	130.0	53.4	47.7	51.4	53.0	47.9	51.1
AssembleCost_NonPr + 1	**A19**	108.5	137.3	130.1	53.3	47.6	51.4	53.8	47.8	51.0
[All]CostPrintable + 1	**A20**	98.8	123.9	84.8	48.2	39.8	8.8	49.6	38.7	8.7
[All]CostPrintable + 2	**A21**	73.5	95.0	57.8	34.4	28.9	6.8	38.2	26.0	6.7
Print & Assemble Pr + 2	**A22**	99.1	136.7	105.8	47.7	42.0	8.7	50.2	41.8	8.7
Base & Print Pr + 2	**A23**	84.7	103.3	64.1	43.1	33.3	7.7	40.5	31.1	7.5
Base & Assemble Pr + 2	**A24**	85.1	101.9	65.0	38.3	30.5	7.8	45.7	34.9	7.6
[All]CostPrintable - 1	**A25**	135.7	164.5	148.8	67.2	57.2	65.3	66.6	57.3	65.3
AssembleCost_Pr - 1	**A26**	135.6	164.1	149.3	68.7	56.9	66.1	65.4	57.0	66.2
PrintCost_Pr - 1	**A27**	137.3	165.5	150.0	66.6	57.1	66.0	69.1	57.3	65.6
BaseCost_Pr - 1	**A28**	134.7	164.5	173.0	66.9	57.1	54.7	66.2	57.0	54.7
Print_Efficiency = 0.25	**B1**	51.1	68.8	65.2	23.5	18.4	12.4	26.7	19.8	11.9
Print_Efficiency = 0.5	**B2**	83.7	106.1	91.4	39.9	33.8	24.5	42.7	33.3	24.0
Print_Efficiency = 1.5	**B3**	136.9	165.6	155.2	68.0	57.5	63.7	67.4	57.3	63.7
Collect_Amount = 0.25	**B4**	96.4	145.8	113.7	48.6	47.7	44.6	46.7	45.7	44.4
Collect_Amount = 0.5	**B5**	130.2	160.3	139.9	64.9	53.6	54.5	64.0	52.9	54.3
Collect_Amount = 1.5	**B6**	140.0	168.5	152.3	69.5	57.5	54.8	69.0	56.8	55.0
Print_Amount = 0.25	**B7**	82.7	84.0	67.7	37.2	22.1	12.6	43.5	22.7	12.2

(continues)

Table B.1. Cycle decision-making algorithm results for time-step 70 (continued).

Cycle Algorithm	ID	Collection Potential			Assembly Potential			Print Potential		
Time-Step: 70		DHE	HHE	HHO	DHE	HHE	HHO	DHE	HHE	HHO
Print_Amount = 0.5	**B8**	137.9	172.2	93.2	64.5	54.3	24.8	71.3	52.8	24.4
Print_Amount = 1.5	**B9**	135.6	165.0	155.4	68.5	57.5	63.9	65.8	56.1	64.2
Collect & Print Amount = 0.5	**B10**	128.4	162.2	84.4	60.2	50.6	25.8	66.2	51.1	25.6
QualityThreshold + 0.1	**C1**	131.4	159.5	137.2	64.6	54.7	49.3	65.3	54.7	49.4
QualityThreshold + 0.2	**C2**	123.7	149.2	122.6	60.8	51.0	44.2	61.4	51.0	44.4
QualityThreshold + 0.3	**C3**	109.2	132.6	95.7	54.9	45.8	32.4	53.1	44.6	32.6
QualityThreshold + 0.4	**C4**	90.2	106.0	60.6	45.0	35.6	18.2	44.3	35.8	18.2
RiskAmount_Print = 1%	**C5**	137.3	167.1	137.7	67.9	56.6	48.6	61.0	47.9	40.8
RiskAmount_Assemble = 1%	**C6**	130.2	154.4	136.3	58.4	45.1	42.9	63.8	53.1	49.4
RiskAmount Pr & As = 10%	**C7**	21.9	23.1	18.6	2.7	1.5	1.4	1.7	1.4	2.1
RiskAmount Pr & As = 15%	**C8**	10.9	15.8	10.9	0.6	0.7	0.6	0.7	0.8	1.0
RiskAmount_Assemble = 15%	**C9**	13.5	13.7	14.1	0.7	0.5	0.6	7.4	6.4	7.3
Quality_incr_Chance = 0.01%	**C10**	134.9	164.5	146.9	66.7	56.8	53.9	66.7	56.1	53.7
Quality_decr_Chance = 25%	**C11**	143.2	173.0	156.1	71.4	59.4	56.7	70.6	58.9	56.8
Quality_decr_Chance = 75%	**C12**	126.3	152.8	139.6	62.6	53.3	51.4	62.1	52.4	51.2
Quality_decr_Upper = 0.5	**C13**	115.7	139.3	106.8	57.3	48.0	38.9	56.8	47.2	38.8
Qual_incr Chance & Upper * 2	**C14**	137.8	166.6	149.8	68.6	57.4	55.4	67.7	56.9	55.2
RiskQuality_Modifier = 10.0	**C15**	133.1	162.1	143.6	65.1	56.3	53.5	66.0	55.8	53.2
RiskQuality_Modifier = 25.0	**C16**	120.8	148.6	131.7	58.9	52.6	50.8	59.4	52.1	50.6
RiskFactory_Modifier = 0.5	**C17**	136.8	164.7	148.5	68.2	56.6	54.3	67.2	56.5	54.3
RiskFactory_Modifier = 1.0	**C18**	137.3	161.1	145.3	68.1	55.3	53.2	67.6	55.0	53.5
Quality Thres & Chance	**C19**	41.6	47.9	32.7	20.6	16.4	8.1	20.4	15.9	8.2
Initial_Printable / 2.0	**D1**	138.7	167.9	90.9	68.3	55.9	15.3	69.0	55.9	15.4
Initial_Printable * 2.0	**D2**	137.2	166.3	149.9	68.1	57.7	66.3	67.6	57.5	66.2
Initial_Materials = 0	**D3**	137.5	169.5	149.1	67.8	57.4	54.4	68.3	57.0	54.5
Initial_Materials / 2.0	**D4**	137.9	167.5	146.2	68.4	57.6	53.9	68.1	56.4	53.6
Initial_Materials * 2.0	**D5**	136.9	165.1	151.6	67.5	56.9	55.2	68.0	56.7	54.9
Env_Materials / 2.0	**D6**	93.4	113.5	109.9	46.6	39.3	41.9	45.8	38.9	41.7
Env_Materials * 2.0	**D7**	127.7	159.0	138.6	59.6	53.7	53.6	66.2	56.8	53.5
Env_Materials * 100	**D8**	114.3	131.5	128.5	48.2	39.7	54.7	63.1	55.0	54.3
Initial_NonPr / 2.0	**D9**	65.5	80.2	78.7	30.6	27.5	31.1	33.8	29.1	30.7
Initial_NonPr * 2.0	**D10**	148.1	180.2	151.3	73.5	61.8	54.3	73.0	61.2	54.2
Initial NonPr & Env * 2.0	**D11**	256.1	312.6	143.3	126.3	106.4	54.6	127.0	106.3	54.5
Initial NonPr & Env * 2.0, Raw=0	**D12**	244.2	299.6	145.2	120.2	100.9	54.4	121.4	100.8	54.5

Table B.2. Cycle decision-making algorithm results for time-step 50.

Cycle Algorithm	ID	Collection Potential			Assembly Potential			Print Potential		
Time-Step: 50		DHE	HHE	HHO	DHE	HHE	HHO	DHE	HHE	HHO
(Default)	**A0**	115.4	156.4	63.1	56.9	52.6	23.7	57.0	52.4	23.7
BaseCost_Pr + 1	**A1**	87.9	114.8	53.6	42.5	36.8	14.1	44.2	36.8	14.0
BaseCost_Pr + 3	**A2**	55.8	60.8	32.5	27.2	19.1	10.9	27.8	20.0	10.8
BaseCost_Pr + 5	**A3**	39.1	37.7	24.4	18.1	10.1	6.0	20.5	12.1	6.0
PrintCost_Pr + 1	**A4**	102.9	146.7	56.8	52.1	47.3	21.8	49.7	47.2	21.7
PrintCost_Pr + 3	**A5**	81.4	119.2	65.9	43.3	38.6	9.8	37.1	35.0	9.8
PrintCost_Pr + 5	**A6**	68.6	101.5	58.6	39.1	32.1	7.8	28.6	27.5	7.8
AssembleCost_Pr + 1	**A7**	101.6	145.6	57.2	48.5	46.7	20.4	51.9	47.5	20.3
AssembleCost_Pr + 3	**A8**	80.1	122.4	65.6	34.9	36.5	9.7	44.3	41.0	9.8
AssembleCost_Pr + 5	**A9**	68.2	107.3	58.3	26.7	29.3	7.7	40.6	36.9	7.8
BaseCost_Time + 2	**A10**	53.6	36.6	33.7	27.8	13.1	13.5	25.3	12.7	13.7
BaseCost_Time - 1	**A11**	139.0	171.7	103.1	68.1	56.9	40.1	69.7	55.6	39.8
PrintCost_Time + 2	**A12**	79.8	102.1	44.8	42.0	35.9	16.3	36.7	31.3	16.4
PrintCost_Time - 1	**A13**	138.3	161.6	82.7	67.8	55.1	32.0	69.4	54.7	32.0
AssembleCost_Time + 2	**A14**	65.3	69.2	44.5	31.4	22.1	16.3	33.0	24.2	16.3
AssembleCost_Time - 1	**A15**	137.2	164.4	81.2	67.8	55.3	32.1	68.0	55.6	32.1
Print & Assemble Time + 2	**A16**	52.6	50.5	31.0	27.4	17.1	12.7	24.6	16.7	12.7
BaseCost_NonPr + 1	**A17**	89.7	100.9	63.3	43.3	35.0	23.6	45.2	35.6	23.7
PrintCost_NonPr + 1	**A18**	107.7	135.2	64.1	53.0	46.8	23.5	53.6	46.6	23.6
AssembleCost_NonPr + 1	**A19**	107.0	135.5	64.4	52.2	46.8	24.2	53.3	46.7	24.3
[All]CostPrintable + 1	**A20**	70.0	92.9	50.7	34.1	28.9	8.8	35.0	28.3	8.8
[All]CostPrintable + 2	**A21**	45.4	57.4	35.7	21.2	17.9	6.8	23.4	15.6	6.7
Print & Assemble Pr + 2	**A22**	68.8	109.1	61.5	33.1	33.1	8.8	34.6	32.7	8.8
Base & Print Pr + 2	**A23**	55.7	67.6	39.7	28.0	21.4	7.7	26.8	20.9	7.8
Base & Assemble Pr + 2	**A24**	56.8	69.5	39.0	25.3	20.7	7.8	30.8	23.7	7.7
[All]CostPrintable - 1	**A25**	135.3	157.4	105.9	66.8	53.7	41.0	66.9	53.8	41.4
AssembleCost_Pr - 1	**A26**	133.8	158.0	67.7	68.2	54.0	28.3	64.0	53.8	28.5
PrintCost_Pr - 1	**A27**	133.5	155.6	66.9	64.8	53.2	28.6	67.3	52.6	28.6
BaseCost_Pr - 1	**A28**	136.4	156.0	94.7	68.1	53.4	25.4	66.9	53.1	25.5
Print_Efficiency = 0.25	**B1**	40.6	53.9	43.0	18.7	14.3	12.6	21.3	16.9	12.6
Print_Efficiency = 0.5	**B2**	63.9	83.9	50.8	30.2	26.6	16.5	32.7	26.4	16.3
Print_Efficiency = 1.5	**B3**	135.8	158.8	76.2	67.8	54.5	27.0	66.3	53.8	27.3
Collect_Amount = 0.25	**B4**	58.4	78.7	61.7	29.8	26.0	23.6	28.2	24.1	23.6
Collect_Amount = 0.5	**B5**	82.5	116.4	62.8	40.9	37.9	23.5	40.6	37.2	23.7
Collect_Amount = 1.5	**B6**	138.4	160.2	65.9	68.9	53.5	24.0	68.2	53.0	24.1
Print_Amount = 0.25	**B7**	49.2	53.3	43.1	22.1	14.1	12.7	26.4	17.0	12.5
Print_Amount = 0.5	**B8**	86.3	84.8	51.5	40.0	27.0	15.9	44.8	26.8	15.8
Print_Amount = 1.5	**B9**	132.9	150.0	76.2	67.1	51.4	27.2	64.6	49.5	27.5
Collect & Print Amount = 0.5	**B10**	69.1	83.0	51.1	32.4	26.4	16.0	35.9	27.2	15.9
QualityThreshold + 0.1	**C1**	109.9	136.3	60.9	54.2	45.9	22.6	54.4	45.5	22.8
QualityThreshold + 0.2	**C2**	95.9	112.8	55.1	47.8	37.7	20.9	46.9	37.2	21.0
QualityThreshold + 0.3	**C3**	71.1	86.8	43.8	35.5	29.3	16.1	34.7	28.6	16.2
QualityThreshold + 0.4	**C4**	46.4	55.5	31.9	23.5	19.1	10.9	22.4	18.1	10.9
RiskAmount_Print = 1%	**C5**	119.0	145.1	61.3	58.9	47.6	23.3	52.3	40.6	20.9
RiskAmount_Assemble = 1%	**C6**	98.8	117.8	57.9	43.6	33.2	19.9	48.5	39.0	22.7

(continues)

Table B.2. Cycle decision-making algorithm results for time-step 50 (continued).

Cycle Algorithm	ID	Collection Potential			Assembly Potential			Print Potential		
Time-Step: 50		**DHE**	**HHE**	**HHO**	**DHE**	**HHE**	**HHO**	**DHE**	**HHE**	**HHO**
RiskAmount Pr & As = 10%	**C7**	17.9	18.0	17.0	2.3	1.8	2.1	2.4	1.6	3.3
RiskAmount Pr & As = 15%	**C8**	10.9	11.4	10.5	1.1	0.8	0.9	1.2	0.9	1.7
RiskAmount_Assemble = 15%	**C9**	13.7	12.8	11.9	1.2	0.6	1.0	7.4	5.7	6.1
Quality_incr_Chance = 0.01%	**C10**	115.3	151.3	63.4	56.8	50.7	23.9	56.9	50.6	23.9
Quality_decr_Chance = 25%	**C11**	124.6	166.0	68.2	61.6	55.9	24.0	61.7	55.9	24.1
Quality_decr_Chance = 75%	**C12**	104.0	138.2	58.8	51.4	46.7	22.3	51.1	46.5	22.3
Quality_decr_Upper = 0.5	**C13**	85.7	110.5	50.1	42.1	37.0	19.7	42.3	36.7	19.7
Qual_incr Chance & Upper * 2	**C14**	120.8	156.6	64.7	60.0	52.6	23.8	59.5	52.3	23.8
RiskQuality_Modifier = 10.0	**C15**	110.6	148.1	59.6	54.7	50.2	23.6	54.2	49.6	23.6
RiskQuality_Modifier = 25.0	**C16**	91.0	128.2	53.1	44.5	44.1	23.1	44.8	43.4	23.1
RiskFactory_Modifier = 0.5	**C17**	119.9	156.5	63.1	59.5	52.5	23.8	59.1	52.2	23.9
RiskFactory_Modifier = 1.0	**C18**	114.6	154.8	63.6	56.7	51.9	23.7	56.5	51.6	23.8
Quality Thres & Chance	**C19**	26.4	24.2	19.6	13.7	8.5	7.3	12.4	8.3	7.3
Initial_Printable / 2.0	**D1**	97.0	141.0	52.6	47.6	45.5	9.5	48.2	45.0	9.4
Initial_Printable * 2.0	**D2**	134.6	155.1	86.5	66.7	52.8	36.5	66.4	52.5	36.5
Initial_Materials = 0	**D3**	103.8	149.6	64.5	51.2	49.2	24.0	51.1	48.9	24.0
Initial_Materials / 2.0	**D4**	111.8	152.9	64.6	55.5	51.4	24.1	55.0	50.2	24.1
Initial_Materials * 2.0	**D5**	125.0	153.4	64.5	61.3	52.0	24.2	62.1	51.8	24.2
Env_Materials / 2.0	**D6**	94.4	114.1	68.2	47.2	39.6	23.4	46.2	38.9	23.4
Env_Materials * 2.0	**D7**	117.7	152.3	63.1	58.3	51.7	24.0	58.0	51.4	24.0
Env_Materials * 100	**D8**	117.4	151.1	62.1	58.1	51.0	23.6	58.1	51.5	23.7
Initial_NonPr / 2.0	**D9**	66.1	80.5	64.1	31.1	27.6	23.8	33.9	29.4	24.0
Initial_NonPr * 2.0	**D10**	115.0	154.5	63.9	57.0	50.8	24.1	56.5	50.6	24.2
Initial NonPr & Env * 2.0	**D11**	118.0	155.0	63.4	59.0	51.3	23.9	57.5	50.6	24.0
Initial NonPr & Env * 2.0, Raw=0	**D12**	103.1	150.5	62.9	50.9	49.4	24.0	50.8	48.8	23.9

Table B.3. Cycle decision-making algorithm results for time-step 30.

Cycle Algorithm	ID	Collection Potential			Assembly Potential			Print Potential		
Time-Step: 30		DHE	HHE	HHO	DHE	HHE	HHO	DHE	HHE	HHO
(Default)	**A0**	38.6	32.0	25.5	20.2	10.8	11.6	18.2	10.5	11.7
BaseCost_Pr + 1	**A1**	33.7	31.3	22.7	17.1	10.7	10.7	16.3	10.4	10.8
BaseCost_Pr + 3	**A2**	25.4	28.1	19.2	12.9	9.2	7.9	12.3	9.5	7.9
BaseCost_Pr + 5	**A3**	20.2	21.8	15.3	10.0	6.5	6.0	10.1	7.4	5.9
PrintCost_Pr + 1	**A4**	37.3	31.5	24.8	19.1	10.6	11.8	17.9	10.5	11.8
PrintCost_Pr + 3	**A5**	30.5	31.2	24.4	15.3	10.5	9.8	14.9	10.3	9.9
PrintCost_Pr + 5	**A6**	28.3	31.9	25.4	15.8	10.5	7.9	12.2	10.4	8.0
AssembleCost_Pr + 1	**A7**	37.6	31.5	24.8	19.0	10.7	11.7	18.3	10.5	11.7
AssembleCost_Pr + 3	**A8**	31.3	31.9	24.5	15.2	10.8	9.8	15.9	10.6	9.8
AssembleCost_Pr + 5	**A9**	27.8	32.2	25.2	12.2	10.5	7.8	15.4	10.5	7.9
BaseCost_Time + 2	**A10**	11.5	13.2	10.1	5.8	4.8	4.8	5.5	4.8	4.9
BaseCost_Time - 1	**A11**	59.0	79.7	37.1	29.8	26.5	14.7	28.8	24.9	14.8
PrintCost_Time + 2	**A12**	24.1	24.9	13.7	13.1	9.1	6.8	10.8	7.6	6.9
PrintCost_Time - 1	**A13**	45.6	41.2	30.2	22.7	13.2	12.7	22.5	14.1	12.8
AssembleCost_Time + 2	**A14**	16.4	19.0	13.6	7.8	5.9	6.7	8.4	6.9	6.9
AssembleCost_Time - 1	**A15**	48.9	51.6	29.7	25.3	18.2	12.7	23.1	16.1	12.7
Print & Assemble Time + 2	**A16**	11.6	15.8	9.3	5.8	5.8	3.9	5.6	5.8	4.0
BaseCost_NonPr + 1	**A17**	37.5	31.6	25.1	19.4	10.7	11.6	17.9	10.5	11.7
PrintCost_NonPr + 1	**A18**	38.5	31.4	25.5	19.9	10.6	11.7	18.3	10.4	11.7
AssembleCost_NonPr + 1	**A19**	37.8	30.8	25.6	19.5	10.5	11.6	18.0	10.3	11.7
[All]CostPrintable + 1	**A20**	28.2	32.4	22.7	14.2	10.6	8.8	13.8	10.5	8.8
[All]CostPrintable + 2	**A21**	22.9	28.5	20.1	11.5	9.2	6.9	11.2	9.4	6.9
Print & Assemble Pr + 2	**A22**	28.3	31.0	25.2	14.1	10.5	8.8	13.9	10.3	8.8
Base & Print Pr + 2	**A23**	25.0	30.2	19.9	12.9	11.2	7.8	11.8	9.6	7.9
Base & Assemble Pr + 2	**A24**	25.3	29.9	20.0	12.0	11.0	7.9	13.2	9.6	7.9
[All]CostPrintable - 1	**A25**	38.9	31.6	23.8	20.3	10.6	11.7	18.3	10.5	11.9
AssembleCost_Pr - 1	**A26**	38.5	31.8	24.8	20.0	10.7	11.7	18.2	10.6	11.8
PrintCost_Pr - 1	**A27**	38.2	31.2	24.3	19.8	10.5	11.4	18.1	10.3	11.6
BaseCost_Pr - 1	**A28**	37.9	31.0	23.7	19.7	10.5	11.6	17.9	10.3	11.7
Print_Efficiency = 0.25	**B1**	29.3	32.0	23.8	15.1	10.5	11.5	14.0	10.4	11.7
Print_Efficiency = 0.5	**B2**	33.5	31.5	24.6	17.2	10.6	11.6	16.1	10.5	11.7
Print_Efficiency = 1.5	**B3**	38.4	31.5	23.6	19.9	10.7	11.6	18.2	10.5	11.7
Collect_Amount = 0.25	**B4**	38.0	30.9	25.5	19.9	10.6	11.6	17.9	9.9	11.7
Collect_Amount = 0.5	**B5**	37.4	31.1	25.4	19.5	10.6	11.6	17.6	10.2	11.7
Collect_Amount = 1.5	**B6**	38.6	31.1	25.5	19.9	10.5	11.6	18.4	10.4	11.7
Print_Amount = 0.25	**B7**	31.3	31.7	23.9	14.9	10.3	11.6	16.1	10.5	11.7
Print_Amount = 0.5	**B8**	37.0	31.7	24.6	18.0	10.8	11.7	18.7	10.5	11.8
Print_Amount = 1.5	**B9**	37.9	31.2	23.7	19.8	10.6	11.6	17.8	10.2	11.7
Collect & Print Amount = 0.5	**B10**	37.1	31.1	24.5	18.3	10.5	11.6	18.5	10.4	11.7
QualityThreshold + 0.1	**C1**	36.2	30.4	24.6	18.9	10.3	11.4	17.1	10.2	11.5
QualityThreshold + 0.2	**C2**	33.8	28.3	22.2	17.5	9.7	10.4	15.9	9.2	10.5
QualityThreshold + 0.3	**C3**	24.0	23.8	17.8	12.2	8.4	8.4	11.5	8.0	8.4
QualityThreshold + 0.4	**C4**	18.8	16.2	12.1	9.7	5.8	5.9	9.0	5.7	5.9
RiskAmount_Print = 1%	**C5**	38.2	31.5	25.2	19.7	10.6	11.6	17.3	9.5	11.5
RiskAmount_Assemble = 1%	**C6**	32.6	27.0	22.6	15.3	8.2	9.7	15.7	9.3	10.5

(continues)

Table B.3. Cycle decision-making algorithm results for time-step 30 (continued).

Cycle Algorithm	ID	Collection Potential			Assembly Potential			Print Potential		
Time-Step: 30		**DHE**	**HHE**	**HHO**	**DHE**	**HHE**	**HHO**	**DHE**	**HHE**	**HHO**
RiskAmount Pr & As = 10%	**C7**	11.2	12.7	9.6	2.6	1.8	1.9	2.3	1.8	3.1
RiskAmount Pr & As = 15%	**C8**	8.5	9.2	8.0	1.4	1.2	1.3	1.2	1.1	2.5
RiskAmount_Assemble = 15%	**C9**	8.8	9.6	7.6	1.3	1.1	1.2	4.3	4.0	3.9
Quality_incr_Chance = 0.01%	**C10**	37.9	31.4	25.3	19.7	10.7	11.5	17.9	10.4	11.7
Quality_decr_Chance = 25%	**C11**	38.8	32.4	26.0	20.1	10.8	11.8	18.5	10.7	11.8
Quality_decr_Chance = 75%	**C12**	36.3	29.9	24.5	19.0	10.3	11.3	17.1	9.8	11.4
Quality_decr_Upper = 0.5	**C13**	30.7	27.0	20.5	15.9	9.4	9.8	14.5	9.0	9.9
Qual_incr Chance & Upper * 2	**C14**	38.0	32.0	25.6	19.6	10.8	11.7	18.1	10.6	11.8
RiskQuality_Modifier = 10.0	**C15**	37.7	30.9	25.1	19.8	10.6	11.6	17.6	10.3	11.7
RiskQuality_Modifier = 25.0	**C16**	35.3	29.0	22.7	18.6	10.3	11.0	16.2	9.9	11.3
RiskFactory_Modifier = 0.5	**C17**	38.4	31.3	25.3	19.9	10.6	11.5	18.2	10.5	11.6
RiskFactory_Modifier = 1.0	**C18**	38.2	31.2	25.5	19.8	10.6	11.5	18.1	10.4	11.7
Quality Thres & Chance	**C19**	10.3	10.2	8.7	5.1	3.7	4.8	5.2	3.7	4.8
Initial_Printable / 2.0	**D1**	30.0	31.5	21.8	15.1	10.7	7.9	14.6	10.5	7.9
Initial_Printable * 2.0	**D2**	38.3	31.8	24.0	20.1	10.8	11.9	17.9	10.5	12.0
Initial_Materials = 0	**D3**	34.0	31.1	25.4	17.5	10.7	11.4	16.3	10.1	11.6
Initial_Materials / 2.0	**D4**	38.1	31.2	25.7	19.7	10.7	11.6	18.2	10.3	11.7
Initial_Materials * 2.0	**D5**	38.8	31.3	25.6	20.0	10.6	11.6	18.5	10.4	11.8
Env_Materials / 2.0	**D6**	38.3	31.4	25.7	19.9	10.6	11.6	18.0	10.4	11.8
Env_Materials * 2.0	**D7**	38.7	32.0	25.4	20.2	10.8	11.7	18.4	10.5	11.7
Env_Materials * 100	**D8**	38.0	31.7	25.8	19.6	10.7	11.8	18.0	10.6	11.8
Initial_NonPr / 2.0	**D9**	38.5	31.6	25.4	19.9	10.6	11.6	18.3	10.3	11.7
Initial_NonPr * 2.0	**D10**	37.9	31.3	25.5	19.7	10.5	11.6	17.9	10.4	11.7
Initial NonPr & Env * 2.0	**D11**	37.7	31.7	25.3	19.5	10.6	11.5	17.8	10.5	11.6
Initial NonPr & Env * 2.0, Raw=0	**D12**	34.0	31.4	25.9	17.5	10.7	11.7	16.2	10.2	11.8

B.2. Results – Variable

Table B.4 lists the results for time-step 40. Table B.5 lists the results for time-step 50. Table B.6 lists the results for time-step 70.

Table B.4. Variable decision-making algorithm results for time-step 40.

Variable Algorithm	ID	Collection Potential			Assembly Potential			Print Potential		
Time-Step: 40		VAR-A	VAR-C	VAR-P	VAR-A	VAR-C	VAR-P	VAR-A	VAR-C	VAR-P
(Default)	**A0**	71.9	95.0	72.4	39.3	36.4	38.5	31.9	29.8	33.1
BaseCost_Pr + 1	**A1**	57.9	68.1	54.3	33.2	26.1	27.4	24.0	22.2	26.2
BaseCost_Pr + 3	**A2**	36.8	41.5	24.3	23.8	14.4	20.4	12.5	13.6	19.9
BaseCost_Pr + 5	**A3**	24.4	27.0	19.2	18.3	9.3	16.3	5.8	10.4	15.8
PrintCost_Pr + 1	**A4**	65.2	85.1	60.5	37.5	33.6	30.9	27.2	26.2	29.0
PrintCost_Pr + 3	**A5**	55.3	70.5	49.9	32.6	28.1	26.4	22.0	21.7	22.9
PrintCost_Pr + 5	**A6**	59.3	64.3	23.7	36.6	28.0	22.2	11.5	12.8	14.9
AssembleCost_Pr + 1	**A7**	71.3	83.5	67.8	33.1	32.3	29.0	26.7	27.1	27.5
AssembleCost_Pr + 3	**A8**	54.8	62.2	49.7	25.0	24.0	22.7	20.5	19.3	26.3
AssembleCost_Pr + 5	**A9**	38.0	55.9	44.3	22.5	20.0	18.8	15.1	17.1	24.9
BaseCost_Time + 2	**A10**	31.2	27.2	16.1	17.2	14.4	14.8	13.6	8.6	15.1
BaseCost_Time - 1	**A11**	118.0	160.6	112.5	62.7	52.0	55.6	54.1	50.6	55.7
PrintCost_Time + 2	**A12**	57.9	73.8	57.1	35.2	31.8	33.4	22.1	20.9	23.1
PrintCost_Time - 1	**A13**	86.1	107.4	47.3	46.5	42.8	36.1	38.9	35.5	37.2
AssembleCost_Time + 2	**A14**	42.4	47.3	36.2	23.2	19.2	18.2	18.9	15.9	17.6
AssembleCost_Time - 1	**A15**	101.7	128.2	88.5	57.0	49.7	45.3	43.8	39.7	42.2
Print & Assemble Time + 2	**A16**	30.6	36.3	25.3	17.0	15.5	10.5	13.2	10.5	9.5
BaseCost_NonPr + 1	**A17**	72.4	93.0	66.8	39.6	35.7	33.7	32.0	29.4	32.4
PrintCost_NonPr + 1	**A18**	72.5	94.8	72.8	39.8	36.5	38.4	32.0	29.7	33.7
AssembleCost_NonPr + 1	**A19**	72.2	95.6	71.5	39.6	35.0	37.7	31.8	29.7	33.1
[All]CostPrintable + 1	**A20**	46.7	53.9	44.4	27.1	21.8	22.0	19.1	13.4	22.0
[All]CostPrintable + 2	**A21**	28.9	39.8	32.1	21.8	13.0	15.7	6.8	11.6	16.0
Print & Assemble Pr + 2	**A22**	47.3	64.1	44.5	27.2	22.7	22.1	19.6	11.9	21.9
Base & Print Pr + 2	**A23**	39.4	41.6	23.2	30.0	19.8	20.2	9.1	9.8	17.3
Base & Assemble Pr + 2	**A24**	34.6	45.8	38.6	23.6	14.7	18.0	10.6	15.6	20.0
[All]CostPrintable - 1	**A25**	97.9	128.0	98.5	66.0	54.7	66.2	59.2	46.8	59.5
AssembleCost_Pr - 1	**A26**	83.2	94.5	81.9	45.3	42.9	43.8	36.9	34.6	37.2
PrintCost_Pr - 1	**A27**	81.0	90.3	80.6	42.8	39.7	42.6	37.4	34.7	37.3
BaseCost_Pr - 1	**A28**	93.5	112.6	93.5	50.2	48.1	49.0	42.2	40.9	43.6
Print_Efficiency = 0.25	**B1**	34.3	42.7	21.6	21.6	14.8	19.2	12.0	11.2	18.9
Print_Efficiency = 0.5	**B2**	48.2	55.8	45.9	26.1	23.6	23.0	21.6	15.0	22.3
Print_Efficiency = 1.5	**B3**	92.7	108.2	91.7	49.9	46.9	47.9	41.8	39.5	42.9
Collect_Amount = 0.25	**B4**	47.0	62.6	45.4	27.9	24.7	23.7	18.6	17.8	21.4
Collect_Amount = 0.5	**B5**	65.8	76.5	55.4	30.3	31.4	28.2	24.2	22.5	26.7
Collect_Amount = 1.5	**B6**	80.7	94.7	74.1	44.1	36.4	37.3	35.8	29.7	36.3
Print_Amount = 0.25	**B7**	36.0	42.3	34.8	23.1	14.7	17.5	12.5	11.2	19.7
Print_Amount = 0.5	**B8**	56.7	51.0	54.0	29.7	24.5	25.5	26.3	17.1	27.6
Print_Amount = 1.5	**B9**	77.4	104.5	77.5	42.1	40.4	42.3	34.7	32.1	34.5

(continues)

Table B.4. Variable decision-making algorithm results for time-step 40 (continued).

Variable Algorithm	ID	Collection Potential			Assembly Potential			Print Potential		
Time-Step: 40		VAR-A	VAR-C	VAR-P	VAR-A	VAR-C	VAR-P	VAR-A	VAR-C	VAR-P
Collect & Print Amount = 0.5	**B10**	52.5	55.4	49.6	28.0	23.6	24.1	23.9	14.8	24.9
QualityThreshold + 0.1	**C1**	69.6	89.3	68.6	38.4	35.1	37.9	30.5	27.9	30.1
QualityThreshold + 0.2	**C2**	63.1	80.5	60.4	34.8	32.1	33.5	27.8	25.0	26.2
QualityThreshold + 0.3	**C3**	48.4	59.6	50.5	27.5	24.9	28.3	20.4	18.4	21.7
QualityThreshold + 0.4	**C4**	34.2	40.6	34.9	19.8	17.5	18.9	14.2	12.8	15.7
RiskAmount_Print = 1%	**C5**	71.4	85.9	72.1	38.8	34.4	38.2	28.5	22.9	29.5
RiskAmount_Assemble = 1%	**C6**	64.5	82.0	63.6	32.1	28.9	32.1	28.5	25.7	27.8
RiskAmount Pr & As = 10%	**C7**	18.5	18.6	16.4	3.6	3.9	4.1	2.1	1.9	2.8
RiskAmount Pr & As = 15%	**C8**	11.6	15.5	12.3	2.1	1.0	2.2	1.8	0.6	1.8
RiskAmount_Assemble = 15%	**C9**	14.2	16.1	13.1	1.8	1.0	1.6	5.4	3.8	4.9
Quality_incr_Chance = 0.01%	**C10**	72.6	92.8	71.4	39.7	35.8	39.1	32.1	29.2	31.5
Quality_decr_Chance = 25%	**C11**	75.3	98.1	75.6	41.3	37.6	40.1	33.3	30.6	34.9
Quality_decr_Chance = 75%	**C12**	67.8	87.9	66.6	37.2	34.5	35.1	29.7	27.9	30.7
Quality_decr_Upper = 0.5	**C13**	58.0	71.1	64.2	32.5	29.1	30.9	24.9	22.3	23.1
Qual_incr Chance & Upper * 2	**C14**	72.9	95.2	72.8	40.3	36.8	40.0	32.0	29.6	32.0
RiskQuality_Modifier = 10.0	**C15**	69.0	90.3	69.2	37.5	35.3	36.5	30.7	28.9	31.8
RiskQuality_Modifier = 25.0	**C16**	71.0	83.5	70.2	32.6	33.0	32.3	25.9	27.5	25.8
RiskFactory_Modifier = 0.5	**C17**	71.6	95.5	72.3	39.3	36.5	39.7	31.5	29.9	31.8
RiskFactory_Modifier = 1.0	**C18**	72.3	94.1	72.2	39.6	36.0	39.6	31.9	29.7	31.8
Quality Thres & Chance	**C19**	19.7	20.7	17.7	11.5	10.0	11.4	8.0	6.7	8.6
Initial_Printable / 2.0	**D1**	58.5	75.4	55.0	34.1	27.4	27.9	23.9	22.7	26.5
Initial_Printable * 2.0	**D2**	95.0	104.8	95.0	50.5	47.9	49.5	43.5	40.8	44.6
Initial_Materials = 0	**D3**	63.9	82.6	59.0	36.4	33.1	29.6	26.8	24.9	28.8
Initial_Materials / 2.0	**D4**	68.5	90.3	63.7	37.9	35.4	32.2	30.0	28.0	30.8
Initial_Materials * 2.0	**D5**	79.1	91.2	79.2	43.4	40.0	41.5	34.7	33.9	36.7
Env_Materials / 2.0	**D6**	72.4	97.4	72.6	39.8	35.9	38.7	32.0	29.7	33.4
Env_Materials * 2.0	**D7**	72.6	91.9	72.0	39.9	35.6	37.9	31.9	29.1	33.2
Env_Materials * 100	**D8**	72.8	93.8	70.8	39.8	36.1	38.6	32.3	29.4	31.3
Initial_NonPr / 2.0	**D9**	58.4	80.8	59.1	37.3	31.1	37.8	33.1	26.9	33.5
Initial_NonPr * 2.0	**D10**	71.5	94.3	73.1	39.3	36.2	40.3	31.5	29.4	32.1
Initial NonPr & Env * 2.0	**D11**	72.0	94.2	66.0	39.5	36.2	33.3	31.7	29.6	32.1
Initial NonPr & Env * 2.0, Raw=0	**D12**	63.5	84.9	58.4	36.2	31.5	29.5	26.7	26.0	28.3

Table B.5. Variable decision-making algorithm results for time-step 50.

Variable Algorithm	ID	Collection Potential			Assembly Potential			Print Potential		
Time-Step: 50		VAR-A	VAR-C	VAR-P	VAR-A	VAR-C	VAR-P	VAR-A	VAR-C	VAR-P
(Default)	**A0**	129.5	166.3	131.4	69.7	63.8	67.0	58.6	53.5	63.0
BaseCost_Pr + 1	**A1**	94.6	121.3	89.0	52.3	42.1	44.1	41.3	38.2	43.7
BaseCost_Pr + 3	**A2**	53.7	60.9	36.1	32.4	19.8	25.9	20.6	20.0	25.6
BaseCost_Pr + 5	**A3**	28.5	36.4	26.7	22.5	14.4	20.2	5.7	15.9	19.0
PrintCost_Pr + 1	**A4**	112.2	155.0	103.2	62.5	57.6	52.1	48.3	47.7	50.0
PrintCost_Pr + 3	**A5**	91.3	123.0	81.1	54.1	47.6	43.1	36.2	35.5	37.2
PrintCost_Pr + 5	**A6**	86.9	95.0	38.4	50.6	38.6	33.3	13.0	18.3	18.3
AssembleCost_Pr + 1	**A7**	138.3	155.3	133.0	55.1	57.7	50.5	48.0	51.4	48.1
AssembleCost_Pr + 3	**A8**	90.0	111.4	80.2	42.0	36.2	35.7	39.3	35.4	43.6
AssembleCost_Pr + 5	**A9**	57.6	94.6	67.2	30.2	28.4	27.5	26.6	30.3	38.9
BaseCost_Time + 2	**A10**	54.8	54.2	27.2	30.6	29.6	25.9	23.5	20.5	26.0
BaseCost_Time - 1	**A11**	139.8	171.0	138.8	74.0	57.1	68.1	64.4	55.5	69.4
PrintCost_Time + 2	**A12**	96.9	133.4	96.7	56.6	52.9	53.3	38.9	37.5	42.2
PrintCost_Time - 1	**A13**	139.1	164.0	86.9	75.7	65.0	54.3	62.1	53.5	57.9
AssembleCost_Time + 2	**A14**	72.3	99.1	67.0	39.7	36.5	32.7	31.8	33.5	33.5
AssembleCost_Time - 1	**A15**	139.0	167.7	134.8	74.8	64.0	66.6	63.0	53.9	66.9
Print & Assemble Time + 2	**A16**	53.9	70.3	42.5	30.1	27.6	19.5	23.2	22.1	18.0
BaseCost_NonPr + 1	**A17**	90.6	102.3	89.3	48.5	40.2	43.5	41.0	33.5	44.6
PrintCost_NonPr + 1	**A18**	112.0	139.0	111.3	60.5	54.0	56.9	50.4	45.7	53.0
AssembleCost_NonPr + 1	**A19**	107.5	138.1	108.8	57.6	50.1	55.1	48.8	44.6	52.4
[All]CostPrintable + 1	**A20**	71.3	81.7	69.8	40.4	29.0	33.8	29.7	21.5	35.1
[All]CostPrintable + 2	**A21**	35.0	57.8	46.7	28.0	17.7	22.4	6.7	15.6	23.7
Print & Assemble Pr + 2	**A22**	72.8	95.3	69.2	41.4	30.0	33.6	30.5	18.6	34.7
Base & Print Pr + 2	**A23**	50.1	57.3	34.3	39.5	23.6	26.2	10.0	13.7	21.7
Base & Assemble Pr + 2	**A24**	46.1	68.0	57.5	30.4	20.9	26.3	15.0	23.0	30.5
[All]CostPrintable - 1	**A25**	111.5	150.2	111.7	79.8	65.3	80.5	75.1	58.9	75.7
AssembleCost_Pr - 1	**A26**	137.3	155.2	138.1	73.9	65.7	71.9	61.7	55.7	64.7
PrintCost_Pr - 1	**A27**	137.1	155.6	139.3	71.5	65.0	72.2	64.0	58.2	65.5
BaseCost_Pr - 1	**A28**	136.9	155.1	137.9	72.1	64.4	70.0	63.2	56.5	66.5
Print_Efficiency = 0.25	**B1**	39.0	51.7	26.2	26.4	15.2	21.1	12.0	12.5	20.7
Print_Efficiency = 0.5	**B2**	68.0	78.2	65.9	36.7	29.7	32.0	30.4	20.4	33.0
Print_Efficiency = 1.5	**B3**	138.2	153.8	136.5	72.6	64.5	68.9	64.2	56.9	66.1
Collect_Amount = 0.25	**B4**	61.6	86.0	58.1	35.7	31.3	29.7	25.3	25.0	28.0
Collect_Amount = 0.5	**B5**	108.5	127.8	84.4	46.1	49.9	42.3	36.2	39.0	41.4
Collect_Amount = 1.5	**B6**	140.5	168.5	136.1	75.1	64.3	67.5	63.9	53.8	67.4
Print_Amount = 0.25	**B7**	40.2	51.5	41.2	27.4	15.1	23.9	12.2	12.6	26.9
Print_Amount = 0.5	**B8**	85.0	69.8	85.1	44.4	33.4	39.8	39.2	27.8	43.7
Print_Amount = 1.5	**B9**	137.0	166.1	136.9	74.0	64.2	73.8	61.7	53.5	62.0
Collect & Print Amount = 0.5	**B10**	73.4	73.9	69.6	39.7	27.7	33.4	32.5	20.7	35.3
QualityThreshold + 0.1	**C1**	122.6	159.3	120.9	66.2	61.4	65.3	55.1	51.3	54.3
QualityThreshold + 0.2	**C2**	105.3	142.1	107.6	57.4	54.9	58.5	46.7	44.8	48.0
QualityThreshold + 0.3	**C3**	81.5	104.4	83.8	45.3	41.0	45.8	35.4	32.2	37.0
QualityThreshold + 0.4	**C4**	54.9	64.6	53.5	30.5	26.9	28.3	23.9	20.0	24.7
RiskAmount_Print = 1%	**C5**	128.4	153.9	129.2	68.8	58.2	66.0	51.7	40.6	54.5
RiskAmount_Assemble = 1%	**C6**	114.2	148.2	112.7	56.2	50.2	55.1	51.2	47.4	50.5

(continues)

Table B.5. Variable decision-making algorithm results for time-step 50 (continued).

Variable Algorithm	ID	Collection Potential			Assembly Potential			Print Potential		
Time-Step: 50		VAR-A	VAR-C	VAR-P	VAR-A	VAR-C	VAR-P	VAR-A	VAR-C	VAR-P
RiskAmount Pr & As = 10%	**C7**	22.6	22.0	18.2	5.0	4.7	3.8	1.8	1.8	2.6
RiskAmount Pr & As = 15%	**C8**	11.9	15.9	12.0	1.6	0.6	1.3	1.4	0.5	1.5
RiskAmount_Assemble = 15%	**C9**	13.6	19.5	15.2	1.0	0.7	1.1	5.5	3.8	6.1
Quality_incr_Chance = 0.01%	**C10**	128.0	162.9	127.1	69.2	62.6	68.4	57.3	53.0	57.1
Quality_decr_Chance = 25%	**C11**	137.8	170.6	137.5	74.3	65.0	70.4	62.4	54.4	65.9
Quality_decr_Chance = 75%	**C12**	114.8	155.5	116.0	61.6	60.5	59.3	51.8	50.5	55.1
Quality_decr_Upper = 0.5	**C13**	95.5	125.8	117.9	51.6	49.0	49.8	42.4	39.3	40.3
Qual_incr Chance & Upper * 2	**C14**	130.0	167.4	132.3	69.9	64.0	71.3	58.7	53.7	59.8
RiskQuality_Modifier = 10.0	**C15**	123.1	161.2	122.2	65.8	62.0	61.4	55.6	52.7	59.0
RiskQuality_Modifier = 25.0	**C16**	129.9	143.2	132.1	52.5	55.1	53.8	45.8	47.9	46.3
RiskFactory_Modifier = 0.5	**C17**	128.9	165.6	129.3	69.6	63.3	69.5	57.9	53.3	58.4
RiskFactory_Modifier = 1.0	**C18**	128.1	164.0	128.7	68.5	63.2	68.8	58.1	53.1	58.3
Quality Thres & Chance	**C19**	28.6	33.5	22.0	16.6	15.2	15.8	11.5	11.0	13.0
Initial_Printable / 2.0	**D1**	103.5	145.7	96.8	57.7	48.3	47.9	44.5	43.4	47.6
Initial_Printable * 2.0	**D2**	137.3	152.6	136.5	72.0	66.1	69.0	63.9	57.9	65.7
Initial_Materials = 0	**D3**	111.7	158.0	106.9	61.2	60.0	53.1	49.0	49.3	52.6
Initial_Materials / 2.0	**D4**	121.4	162.4	114.0	65.0	62.4	56.7	55.0	51.9	55.9
Initial_Materials * 2.0	**D5**	137.4	155.8	136.3	73.2	64.5	69.1	62.7	55.9	65.8
Env_Materials / 2.0	**D6**	95.3	114.3	95.3	52.7	42.3	49.6	41.8	36.6	44.8
Env_Materials * 2.0	**D7**	127.9	156.7	127.1	68.6	60.0	64.4	57.7	53.3	61.3
Env_Materials * 100	**D8**	128.6	157.6	127.0	68.8	60.4	67.6	58.3	53.0	58.0
Initial_NonPr / 2.0	**D9**	58.1	80.7	57.8	36.9	31.3	36.9	35.0	26.7	35.0
Initial_NonPr * 2.0	**D10**	131.2	175.1	127.6	70.3	65.8	68.6	59.5	56.4	57.5
Initial NonPr & Env * 2.0	**D11**	128.3	176.9	116.8	68.6	64.4	58.0	57.9	56.5	57.4
Initial NonPr & Env * 2.0, Raw=0	**D12**	112.9	160.5	105.2	62.2	54.1	52.4	49.5	50.3	51.6

Table B.6. Variable decision-making algorithm results for time-step 70.

Variable Algorithm	ID	Collection Potential			Assembly Potential			Print Potential		
Time-Step: 70		VAR-A	VAR-C	VAR-P	VAR-A	VAR-C	VAR-P	VAR-A	VAR-C	VAR-P
(Default)	**A0**	138.0	165.6	136.7	74.1	64.0	69.4	62.5	54.0	65.9
BaseCost_Pr + 1	**A1**	117.4	139.9	113.6	67.6	50.0	61.2	54.4	46.0	61.9
BaseCost_Pr + 3	**A2**	83.8	94.3	62.0	50.1	30.9	50.1	32.7	30.6	51.6
BaseCost_Pr + 5	**A3**	39.0	65.3	51.7	33.0	28.0	31.8	5.5	31.2	31.9
PrintCost_Pr + 1	**A4**	136.3	167.6	134.0	75.8	63.4	66.9	59.1	52.0	63.8
PrintCost_Pr + 3	**A5**	116.2	145.6	111.0	68.9	57.1	59.3	46.0	42.3	50.3
PrintCost_Pr + 5	**A6**	147.0	145.2	81.8	82.9	58.2	58.2	15.4	30.3	35.9
AssembleCost_Pr + 1	**A7**	154.3	161.6	153.0	62.8	60.7	60.4	55.4	54.0	57.4
AssembleCost_Pr + 3	**A8**	116.0	140.2	110.8	54.7	46.0	49.7	52.7	47.3	59.9
AssembleCost_Pr + 5	**A9**	89.3	128.3	97.9	44.5	38.7	40.0	43.7	42.6	56.8
BaseCost_Time + 2	**A10**	136.0	152.3	73.3	74.6	67.8	53.0	59.4	52.1	53.5
BaseCost_Time - 1	**A11**	138.7	167.9	138.1	73.5	56.5	68.1	64.0	54.5	68.5
PrintCost_Time + 2	**A12**	138.8	166.4	138.6	75.9	64.4	71.1	61.4	52.7	65.8
PrintCost_Time - 1	**A13**	138.5	165.0	117.2	75.0	65.5	78.4	62.1	54.0	80.3
AssembleCost_Time + 2	**A14**	137.6	163.5	136.7	76.7	63.8	70.0	59.3	54.1	65.4
AssembleCost_Time - 1	**A15**	140.2	167.0	135.6	75.3	63.2	66.8	63.4	54.0	67.6
Print & Assemble Time + 2	**A16**	136.3	162.8	130.6	74.7	64.2	66.4	59.5	55.2	58.8
BaseCost_NonPr + 1	**A17**	91.0	101.4	90.2	49.0	40.2	43.3	41.0	33.3	45.8
PrintCost_NonPr + 1	**A18**	112.3	139.9	110.7	60.6	54.1	56.3	50.5	45.7	53.1
AssembleCost_NonPr + 1	**A19**	107.2	138.0	107.6	57.7	49.9	54.2	48.3	44.8	52.3
[All]CostPrintable + 1	**A20**	99.4	125.3	98.8	55.8	43.7	48.1	42.4	34.2	49.4
[All]CostPrintable + 2	**A21**	48.6	93.9	72.2	41.5	29.2	35.1	6.4	24.9	36.3
Print & Assemble Pr + 2	**A22**	99.2	150.8	97.8	55.9	45.7	47.7	42.2	33.2	48.9
Base & Print Pr + 2	**A23**	78.8	103.3	74.9	65.8	38.6	46.1	12.1	26.3	41.7
Base & Assemble Pr + 2	**A24**	78.4	100.5	85.1	48.7	30.8	38.1	28.7	34.7	45.9
[All]CostPrintable - 1	**A25**	112.9	150.8	112.2	80.5	65.3	80.1	76.2	58.7	75.7
AssembleCost_Pr - 1	**A26**	138.3	155.3	138.1	74.9	65.8	71.7	61.9	56.0	64.9
PrintCost_Pr - 1	**A27**	137.7	155.3	136.6	71.5	64.1	71.0	64.8	58.3	64.0
BaseCost_Pr - 1	**A28**	136.6	153.5	138.3	72.5	64.0	70.4	62.6	56.2	66.6
Print_Efficiency = 0.25	**B1**	52.1	79.1	32.0	39.6	15.6	26.2	11.8	12.5	26.6
Print_Efficiency = 0.5	**B2**	86.9	107.6	85.7	45.5	38.7	40.4	36.6	29.2	41.9
Print_Efficiency = 1.5	**B3**	136.1	154.9	136.7	71.7	65.4	68.8	62.8	56.9	66.3
Collect_Amount = 0.25	**B4**	98.5	147.9	113.0	57.3	51.3	43.4	40.1	45.2	42.3
Collect_Amount = 0.5	**B5**	146.4	157.8	130.6	65.3	62.0	63.4	53.1	50.3	63.5
Collect_Amount = 1.5	**B6**	140.9	168.6	139.5	75.4	64.9	69.1	64.0	54.2	68.9
Print_Amount = 0.25	**B7**	54.4	76.2	80.6	41.6	19.9	44.8	12.0	15.6	49.9
Print_Amount = 0.5	**B8**	140.3	163.1	138.4	73.7	63.9	65.1	64.5	55.8	71.1
Print_Amount = 1.5	**B9**	137.5	164.8	138.6	74.1	64.1	74.7	62.3	53.2	62.8
Collect & Print Amount = 0.5	**B10**	132.1	153.5	127.9	70.3	50.6	60.6	59.7	43.4	65.5
QualityThreshold + 0.1	**C1**	134.0	162.0	133.8	72.7	62.7	72.1	60.5	51.7	60.4
QualityThreshold + 0.2	**C2**	127.8	148.6	124.6	69.8	58.3	68.9	56.7	47.8	57.3
QualityThreshold + 0.3	**C3**	113.8	135.0	106.1	62.5	53.7	64.4	50.1	42.9	53.2
QualityThreshold + 0.4	**C4**	93.1	114.0	94.8	51.9	45.4	48.3	40.2	36.7	45.6
RiskAmount_Print = 1%	**C5**	139.1	166.0	138.3	74.9	64.1	70.5	56.7	45.4	60.2
RiskAmount_Assemble = 1%	**C6**	129.2	159.0	128.7	64.4	54.9	63.6	59.0	50.8	59.4

(continues)

Table B.6. Variable decision-making algorithm results for time-step 70 (continued).

Variable Algorithm	ID	Collection Potential			Assembly Potential			Print Potential		
Time-Step: 70		VAR-A	VAR-C	VAR-P	VAR-A	VAR-C	VAR-P	VAR-A	VAR-C	VAR-P
RiskAmount Pr & As = 10%	**C7**	23.4	27.7	25.5	5.7	4.0	3.5	1.7	1.1	2.0
RiskAmount Pr & As = 15%	**C8**	13.1	20.1	11.6	2.1	0.5	2.2	0.5	0.3	0.9
RiskAmount_Assemble = 15%	**C9**	17.0	16.6	14.8	0.9	0.5	1.1	7.7	4.4	8.1
Quality_incr_Chance = 0.01%	**C10**	137.4	164.5	138.6	74.0	63.5	74.8	61.8	53.2	62.5
Quality_decr_Chance = 25%	**C11**	144.4	171.6	143.7	78.4	65.9	73.5	64.8	54.4	69.2
Quality_decr_Chance = 75%	**C12**	129.4	157.9	130.2	70.6	61.1	66.4	58.4	51.8	63.4
Quality_decr_Upper = 0.5	**C13**	119.8	142.2	136.1	65.2	56.3	57.9	53.0	46.0	47.8
Qual_incr Chance & Upper * 2	**C14**	139.5	167.0	139.6	75.0	64.2	75.1	63.0	53.8	63.2
RiskQuality_Modifier = 10.0	**C15**	134.4	160.9	133.4	72.3	61.8	67.8	60.7	52.8	64.2
RiskQuality_Modifier = 25.0	**C16**	141.2	149.4	141.9	58.8	58.5	58.5	50.7	51.0	51.4
RiskFactory_Modifier = 0.5	**C17**	138.3	165.3	138.6	74.5	63.7	74.9	62.6	53.4	62.7
RiskFactory_Modifier = 1.0	**C18**	138.4	166.5	138.8	74.9	64.1	74.7	62.4	53.3	62.7
Quality Thres & Chance	**C19**	48.4	55.8	40.7	28.3	25.0	26.0	19.5	16.1	23.7
Initial_Printable / 2.0	**D1**	139.7	172.8	138.8	76.6	59.6	69.3	61.7	54.2	68.1
Initial_Printable * 2.0	**D2**	137.1	151.4	138.1	72.2	65.7	70.1	63.4	57.3	66.6
Initial_Materials = 0	**D3**	138.7	166.9	134.2	75.7	64.3	70.0	61.5	52.6	69.8
Initial_Materials / 2.0	**D4**	138.4	166.0	134.3	74.9	64.2	68.1	62.8	53.2	67.2
Initial_Materials * 2.0	**D5**	139.2	156.5	138.0	74.4	64.9	70.0	63.2	56.5	66.6
Env_Materials / 2.0	**D6**	94.4	114.6	95.4	51.9	42.8	49.9	41.6	36.4	44.7
Env_Materials * 2.0	**D7**	129.1	156.6	129.1	66.3	59.3	61.9	60.8	52.9	65.2
Env_Materials * 100	**D8**	115.0	129.0	116.2	52.5	44.7	53.2	59.4	51.6	60.2
Initial_NonPr / 2.0	**D9**	57.5	79.8	58.8	36.7	30.7	37.6	34.9	26.5	35.2
Initial_NonPr * 2.0	**D10**	150.1	178.3	149.0	80.6	67.8	80.4	69.2	58.4	68.2
Initial NonPr & Env * 2.0	**D11**	260.8	314.4	258.1	136.5	114.7	128.0	121.2	103.9	127.3
Initial NonPr & Env * 2.0, Raw=0	**D12**	250.1	305.6	244.2	132.1	106.9	120.7	115.4	100.0	120.6

B.3. Results – Strategic

Table B.7. lists the results for Strategic-A. Table B.8. lists the results for Strategic-C. Table B.9 lists the results for Strategic-P.

Table B.7. Strategic-A decision-making algorithm results.

Strategic-A	ID	Assembly Potential			Collection Potential			Print Potential		
		30	**50**	**70**	**30**	**50**	**70**	**30**	**50**	**70**
(Default)	**A0**	33.7	91.1	117.0	39.5	113.6	139.9	5.5	21.5	21.4
BaseCost_Pr + 1	**A1**	27.0	64.7	102.7	32.9	83.4	121.9	5.8	17.5	18.1
BaseCost_Pr + 3	**A2**	18.9	40.2	70.6	25.0	54.6	85.3	5.7	14.0	14.3
BaseCost_Pr + 5	**A3**	13.5	26.9	48.9	19.3	38.3	61.1	5.3	11.1	11.6
PrintCost_Pr + 1	**A4**	34.8	85.4	119.6	40.6	105.8	140.5	5.6	19.4	19.4
PrintCost_Pr + 3	**A5**	31.9	71.7	121.7	37.6	88.8	139.4	5.6	16.0	16.4
PrintCost_Pr + 5	**A6**	28.7	68.8	123.5	34.6	83.2	138.0	5.8	13.7	13.6
AssembleCost_Pr + 1	**A7**	27.6	68.9	103.4	33.5	89.0	123.6	5.7	19.4	19.2
AssembleCost_Pr + 3	**A8**	19.6	50.3	74.3	25.3	69.7	93.1	5.5	18.7	17.8
AssembleCost_Pr + 5	**A9**	16.0	37.2	56.6	21.7	56.0	75.2	5.6	18.0	17.8
BaseCost_Time + 2	**A10**	8.0	42.9	95.5	11.8	51.0	112.7	3.8	7.5	15.5
BaseCost_Time - 1	**A11**	41.6	107.8	108.5	50.0	140.8	142.0	8.2	31.8	32.4
PrintCost_Time + 2	**A12**	32.0	65.0	117.7	36.7	81.6	135.1	4.6	16.0	16.1
PrintCost_Time - 1	**A13**	36.0	106.8	116.6	41.9	132.1	141.4	5.6	23.9	23.7
AssembleCost_Time + 2	**A14**	11.2	45.5	90.6	14.9	56.0	101.2	3.6	9.7	9.4
AssembleCost_Time - 1	**A15**	38.3	110.2	112.5	46.1	139.3	141.9	7.3	27.4	28.0
Print & Assemble Time + 2	**A16**	8.0	41.4	75.1	11.9	48.7	83.2	3.9	6.9	7.6
BaseCost_NonPr + 1	**A17**	35.3	67.2	67.5	40.8	89.7	89.7	5.5	21.2	21.5
PrintCost_NonPr + 1	**A18**	35.3	90.6	99.6	40.9	112.9	121.7	5.5	21.3	20.7
AssembleCost_NonPr + 1	**A19**	35.1	73.5	75.8	41.0	95.8	98.7	5.6	21.5	22.1
[All]CostPrintable + 1	**A20**	21.8	49.4	82.0	27.7	66.4	98.5	5.6	16.2	15.1
[All]CostPrintable + 2	**A21**	15.8	33.0	59.8	21.8	46.1	73.0	5.6	12.7	12.7
Print & Assemble Pr + 2	**A22**	21.9	49.0	84.5	27.6	65.6	101.0	5.5	15.7	15.5
Base & Print Pr + 2	**A23**	21.3	42.6	79.9	27.0	56.4	94.0	5.6	13.3	13.2
Base & Assemble Pr + 2	**A24**	16.9	36.5	61.7	22.7	52.5	76.5	5.7	15.3	13.9
[All]CostPrintable - 1	**A25**	45.6	109.7	109.3	51.5	137.2	135.8	5.4	26.2	25.1
AssembleCost_Pr - 1	**A26**	41.7	108.6	112.0	47.3	133.6	137.3	5.2	23.5	24.0
PrintCost_Pr - 1	**A27**	36.5	97.4	115.2	42.4	123.1	141.4	5.7	24.5	24.6
BaseCost_Pr - 1	**A28**	45.0	112.4	111.5	50.7	140.1	137.8	5.4	26.6	24.4
Print_Efficiency = 0.25	**B1**	22.6	24.3	37.1	28.1	39.1	52.2	5.4	14.1	14.4
Print_Efficiency = 0.5	**B2**	26.3	45.1	68.5	32.1	63.2	86.2	5.6	17.4	16.9
Print_Efficiency = 1.5	**B3**	44.0	108.9	111.1	49.9	135.3	138.0	5.7	24.9	25.3
Collect_Amount = 0.25	**B4**	32.7	41.4	82.2	38.3	57.0	96.6	5.3	15.1	13.7
Collect_Amount = 0.5	**B5**	33.5	66.5	113.0	39.4	86.0	132.8	5.6	18.6	18.8
Collect_Amount = 1.5	**B6**	35.7	102.7	119.5	41.7	126.7	143.2	5.5	22.7	22.3
Print_Amount = 0.25	**B7**	23.6	29.5	47.6	29.6	47.0	65.0	5.9	16.9	15.9
Print_Amount = 0.5	**B8**	28.2	51.9	98.9	34.1	72.3	119.5	5.5	19.3	18.9
Print_Amount = 1.5	**B9**	37.2	112.3	117.6	43.0	135.9	140.5	5.5	22.4	22.0

(continues)

Table B.7. Strategic-A decision-making algorithm results (continued).

Strategic-A	ID	Assembly Potential			Collection Potential			Print Potential		
		30	**50**	**70**	**30**	**50**	**70**	**30**	**50**	**70**
Collect & Print Amount = 0.5	**B10**	29.2	48.2	88.6	35.2	67.3	108.3	5.8	18.4	18.3
QualityThreshold + 0.1	**C1**	33.5	84.8	115.3	39.2	106.3	137.1	5.5	20.6	20.5
QualityThreshold + 0.2	**C2**	31.2	74.6	114.0	36.4	93.2	132.4	4.9	18.0	17.0
QualityThreshold + 0.3	**C3**	26.3	62.3	103.2	30.7	78.8	120.7	4.4	16.0	16.4
QualityThreshold + 0.4	**C4**	15.6	38.4	72.5	19.1	50.6	84.6	3.2	11.7	11.7
RiskAmount_Print = 1%	**C5**	35.0	83.4	118.7	40.8	105.4	141.3	4.8	15.9	14.8
RiskAmount_Assemble = 1%	**C6**	28.6	75.9	103.8	35.3	102.1	128.7	5.0	20.2	18.2
RiskAmount Pr & As = 10%	**C7**	6.8	6.5	6.9	14.6	20.3	20.0	1.4	1.7	1.2
RiskAmount Pr & As = 15%	**C8**	2.0	1.5	2.0	9.0	11.5	12.6	1.0	1.1	0.9
RiskAmount_Assemble = 15%	**C9**	3.0	2.4	1.3	10.6	13.8	13.9	3.0	5.2	4.4
Quality_incr_Chance = 0.01%	**C10**	33.7	90.8	116.6	39.4	113.3	139.0	5.3	21.4	21.3
Quality_decr_Chance = 25%	**C11**	35.9	95.4	119.2	41.6	118.9	142.5	5.5	22.6	21.9
Quality_decr_Chance = 75%	**C12**	33.3	86.9	111.4	38.9	109.0	133.0	5.2	21.2	19.0
Quality_decr_Upper = 0.5	**C13**	28.8	70.8	107.3	33.8	90.2	125.0	5.0	18.4	16.3
Qual_incr Chance & Upper * 2	**C14**	35.5	90.1	117.7	41.1	112.6	140.9	5.5	21.5	22.2
RiskQuality_Modifier = 10.0	**C15**	34.4	84.0	113.2	40.3	106.4	134.9	5.4	21.1	20.0
RiskQuality_Modifier = 25.0	**C16**	32.1	66.2	105.3	37.3	84.7	125.3	4.7	17.1	18.2
RiskFactory_Modifier = 0.5	**C17**	35.4	89.8	117.2	41.0	113.0	140.3	5.4	21.9	21.7
RiskFactory_Modifier = 1.0	**C18**	34.4	90.0	117.7	40.2	112.9	140.1	5.4	21.9	21.0
Quality Thres & Chance	**C19**	8.0	17.8	30.4	10.6	23.4	36.7	2.6	5.1	6.0
Initial_Printable / 2.0	**D1**	23.1	76.6	123.6	29.0	95.5	141.5	5.8	18.1	16.5
Initial_Printable * 2.0	**D2**	44.1	111.4	113.0	49.9	140.2	140.2	5.4	27.6	26.1
Initial_Materials = 0	**D3**	27.1	82.3	121.7	32.5	102.2	142.1	5.2	18.7	19.3
Initial_Materials / 2.0	**D4**	31.8	85.7	118.1	37.4	106.9	139.8	5.4	20.3	20.3
Initial_Materials * 2.0	**D5**	39.3	96.2	113.1	45.2	122.0	138.7	5.8	24.4	24.3
Env_Materials / 2.0	**D6**	35.2	72.4	72.6	41.1	95.3	95.8	5.5	22.1	22.4
Env_Materials * 2.0	**D7**	35.1	85.8	108.0	40.9	108.3	130.2	5.6	21.0	20.4
Env_Materials * 100	**D8**	35.4	88.8	88.2	41.2	111.5	111.0	5.6	21.8	21.4
Initial_NonPr / 2.0	**D9**	34.7	42.7	44.0	40.4	65.5	67.0	5.6	21.5	22.1
Initial_NonPr * 2.0	**D10**	35.1	91.1	129.9	41.0	114.1	152.2	5.4	21.5	21.2
Initial NonPr & Env * 2.0	**D11**	33.9	87.8	187.8	39.7	110.2	211.3	5.6	21.4	20.7
Initial NonPr & Env * 2.0, Raw=0	**D12**	26.8	82.8	154.9	31.9	102.9	173.5	4.9	19.6	16.9

Table B.8. Strategic-C decision-making algorithm results.

Strategic-C	ID	Assembly Potential			Collection Potential			Print Potential		
		30	50	70	30	50	70	30	50	70
(Default)	**A0**	6.0	22.5	21.8	44.0	216.7	234.6	6.0	22.3	21.3
BaseCost_Pr + 1	**A1**	5.6	17.1	16.8	42.1	132.6	181.3	5.5	17.7	17.4
BaseCost_Pr + 3	**A2**	5.5	12.3	12.0	33.8	72.0	110.6	5.9	13.9	13.4
BaseCost_Pr + 5	**A3**	5.2	9.5	9.5	24.4	46.9	78.0	5.8	11.3	11.4
PrintCost_Pr + 1	**A4**	6.0	19.9	20.0	45.0	198.5	241.8	5.9	19.8	20.6
PrintCost_Pr + 3	**A5**	5.9	18.1	16.9	44.1	174.8	245.2	5.9	17.0	16.7
PrintCost_Pr + 5	**A6**	5.8	17.0	16.9	44.0	145.1	243.2	5.8	13.5	13.1
AssembleCost_Pr + 1	**A7**	5.9	19.7	19.6	44.0	192.6	244.4	5.9	19.3	20.4
AssembleCost_Pr + 3	**A8**	5.9	16.2	16.2	43.8	166.6	248.3	5.9	17.7	17.0
AssembleCost_Pr + 5	**A9**	6.0	13.5	13.5	44.4	145.2	246.2	5.9	18.1	18.3
BaseCost_Time + 2	**A10**	4.0	8.4	18.6	15.0	44.5	138.5	4.0	7.8	15.6
BaseCost_Time - 1	**A11**	8.4	31.7	31.9	97.1	220.3	220.3	8.5	33.6	33.0
PrintCost_Time + 2	**A12**	4.9	16.6	17.1	36.3	150.9	235.9	5.0	16.2	16.3
PrintCost_Time - 1	**A13**	6.0	24.6	24.5	47.9	233.4	237.2	5.7	24.6	24.2
AssembleCost_Time + 2	**A14**	3.7	12.0	11.8	27.2	111.9	217.2	3.9	10.3	9.7
AssembleCost_Time - 1	**A15**	8.9	26.7	27.2	64.9	224.1	224.3	7.4	27.8	27.5
Print & Assemble Time + 2	**A16**	4.0	8.6	8.1	25.6	84.8	163.2	4.0	8.1	7.9
BaseCost_NonPr + 1	**A17**	5.8	22.2	20.9	43.2	115.0	115.4	5.8	21.6	20.8
PrintCost_NonPr + 1	**A18**	5.9	22.6	22.1	43.4	210.8	215.7	5.6	21.9	22.4
AssembleCost_NonPr + 1	**A19**	5.9	22.7	22.6	44.8	206.6	214.4	5.9	21.6	22.3
[All]CostPrintable + 1	**A20**	6.0	15.4	15.0	44.7	122.8	174.4	5.9	16.3	15.8
[All]CostPrintable + 2	**A21**	5.7	10.2	11.2	37.2	77.5	127.6	6.0	13.0	12.4
Print & Assemble Pr + 2	**A22**	6.0	16.4	15.7	44.4	160.1	246.5	5.9	15.8	15.6
Base & Print Pr + 2	**A23**	5.5	13.0	12.8	37.7	81.4	133.6	5.8	13.2	13.5
Base & Assemble Pr + 2	**A24**	5.9	12.5	11.8	40.0	88.1	133.2	5.8	14.8	15.0
[All]CostPrintable - 1	**A25**	6.0	28.2	29.0	44.2	216.3	222.7	5.7	25.5	25.7
AssembleCost_Pr - 1	**A26**	6.0	25.6	26.4	45.1	227.0	232.0	5.9	25.0	24.5
PrintCost_Pr - 1	**A27**	5.7	25.2	25.5	42.4	223.3	231.7	5.7	24.4	24.9
BaseCost_Pr - 1	**A28**	5.6	29.2	28.6	43.0	221.9	221.0	5.7	26.8	26.2
Print_Efficiency = 0.25	**B1**	5.8	12.0	11.9	42.5	59.0	80.3	5.7	14.9	14.8
Print_Efficiency = 0.5	**B2**	5.8	16.3	16.2	44.1	107.4	143.6	5.9	17.0	17.1
Print_Efficiency = 1.5	**B3**	6.0	28.9	27.7	43.3	224.4	226.4	5.6	25.8	25.9
Collect_Amount = 0.25	**B4**	5.9	17.2	16.2	43.6	114.4	221.5	5.8	16.2	15.8
Collect_Amount = 0.5	**B5**	5.9	19.4	20.0	44.0	168.3	237.5	5.8	19.1	18.8
Collect_Amount = 1.5	**B6**	5.8	23.5	23.9	44.8	231.5	238.3	6.0	22.5	22.7
Print_Amount = 0.25	**B7**	6.0	14.0	13.3	44.0	66.8	106.2	6.0	16.4	15.8
Print_Amount = 0.5	**B8**	6.0	18.2	16.9	44.9	124.5	225.7	5.8	19.9	19.5
Print_Amount = 1.5	**B9**	5.9	24.0	24.0	43.4	222.2	234.0	5.8	22.8	22.1
Collect & Print Amount = 0.5	**B10**	5.9	16.7	15.9	43.5	111.5	200.8	5.8	18.7	18.3
QualityThreshold + 0.1	**C1**	5.7	22.1	21.9	42.0	205.8	231.2	5.7	21.2	21.7
QualityThreshold + 0.2	**C2**	5.6	19.5	20.0	39.8	168.2	216.6	5.6	18.6	17.9
QualityThreshold + 0.3	**C3**	5.0	17.6	15.5	32.1	141.9	184.8	4.7	17.2	14.9
QualityThreshold + 0.4	**C4**	3.8	8.8	11.9	21.4	59.9	141.0	3.2	9.0	11.5
RiskAmount_Print = 1%	**C5**	6.0	22.5	21.1	44.3	199.4	229.2	4.9	16.2	13.8
RiskAmount_Assemble = 1%	**C6**	4.9	14.8	13.1	39.8	166.6	225.0	5.7	19.4	19.2

(continues)

Table B.8. Strategic-C decision-making algorithm results (continued).

Strategic-C	ID	Assembly Potential			Collection Potential			Print Potential		
		30	**50**	**70**	**30**	**50**	**70**	**30**	**50**	**70**
RiskAmount Pr & As = 10%	**C7**	0.9	0.5	0.6	12.2	15.5	25.7	1.0	1.3	1.2
RiskAmount Pr & As = 15%	**C8**	0.9	0.5	0.4	12.5	14.3	16.2	1.0	1.1	1.2
RiskAmount_Assemble = 15%	**C9**	0.8	0.9	0.4	12.2	17.4	17.9	3.1	5.6	5.7
Quality_incr_Chance = 0.01%	**C10**	6.0	23.2	21.4	44.6	219.3	238.8	5.9	21.8	21.7
Quality_decr_Chance = 25%	**C11**	5.8	23.3	23.1	44.4	223.1	245.9	5.9	21.9	22.8
Quality_decr_Chance = 75%	**C12**	5.8	21.4	21.5	42.0	194.6	227.4	5.5	21.4	20.9
Quality_decr_Upper = 0.5	**C13**	5.7	18.9	17.1	37.9	155.4	201.9	5.2	18.0	15.9
Qual_incr Chance & Upper * 2	**C14**	5.7	22.0	22.5	41.5	218.1	240.5	5.8	22.2	21.6
RiskQuality_Modifier = 10.0	**C15**	5.8	21.5	21.1	40.4	205.1	233.8	5.5	20.9	20.1
RiskQuality_Modifier = 25.0	**C16**	5.7	19.1	17.7	38.3	175.3	214.5	5.8	18.4	17.3
RiskFactory_Modifier = 0.5	**C17**	5.7	22.9	22.2	42.8	215.0	240.6	5.7	22.1	21.5
RiskFactory_Modifier = 1.0	**C18**	5.8	22.4	22.4	42.7	212.3	239.9	5.9	21.4	22.8
Quality Thres & Chance	**C19**	2.8	7.7	6.8	13.5	37.9	62.2	2.8	6.3	6.7
Initial_Printable / 2.0	**D1**	5.9	18.2	17.7	44.2	176.7	248.6	5.8	17.7	18.1
Initial_Printable * 2.0	**D2**	5.5	29.0	27.4	42.0	226.8	228.0	5.7	26.0	26.1
Initial_Materials = 0	**D3**	5.9	19.5	19.2	43.7	193.7	248.5	5.7	19.6	17.9
Initial_Materials / 2.0	**D4**	5.9	21.6	20.9	43.7	206.4	241.2	5.8	20.8	20.2
Initial_Materials * 2.0	**D5**	5.7	25.8	25.2	42.2	230.2	234.8	6.0	25.6	24.7
Env_Materials / 2.0	**D6**	6.0	23.5	22.2	44.6	147.2	148.6	5.8	23.1	22.1
Env_Materials * 2.0	**D7**	6.0	22.4	22.4	43.5	206.8	230.7	5.9	21.9	21.3
Env_Materials * 100	**D8**	5.8	21.8	17.6	43.6	194.8	167.6	5.7	21.4	22.0
Initial_NonPr / 2.0	**D9**	5.9	21.8	21.2	43.8	93.1	92.6	5.8	21.2	21.8
Initial_NonPr * 2.0	**D10**	5.8	22.4	22.9	44.2	213.6	265.4	5.9	22.2	22.4
Initial NonPr & Env * 2.0	**D11**	6.0	22.7	22.1	44.9	210.1	411.0	5.7	22.1	20.6
Initial NonPr & Env * 2.0, Raw=0	**D12**	5.9	18.9	18.2	44.0	178.7	355.4	5.6	18.8	18.4

Table B.9. Strategic-P decision-making algorithm results.

Strategic-P	ID	Assembly Potential			Collection Potential			Print Potential		
		30	50	70	30	50	70	30	50	70
(Default)	**A0**	7.8	24.6	24.4	34.0	126.1	139.7	25.9	100.5	114.0
BaseCost_Pr + 1	**A1**	7.7	19.5	20.1	34.3	89.1	121.0	26.1	68.6	99.7
BaseCost_Pr + 3	**A2**	7.2	16.2	16.2	26.2	57.8	86.2	18.9	40.9	69.0
BaseCost_Pr + 5	**A3**	6.7	12.6	11.3	20.5	37.1	65.4	13.8	24.1	53.5
PrintCost_Pr + 1	**A4**	7.7	23.1	22.2	34.1	105.1	124.4	26.3	80.9	100.8
PrintCost_Pr + 3	**A5**	7.5	18.8	19.6	28.2	74.5	96.0	20.6	54.8	75.4
PrintCost_Pr + 5	**A6**	7.1	21.8	21.2	24.3	63.5	80.1	17.1	41.1	58.3
AssembleCost_Pr + 1	**A7**	7.7	22.1	22.2	34.9	114.4	140.2	26.9	90.7	117.0
AssembleCost_Pr + 3	**A8**	8.0	17.5	16.9	35.1	93.9	137.9	27.1	75.6	119.8
AssembleCost_Pr + 5	**A9**	7.7	15.3	14.9	34.2	76.0	127.6	26.3	59.7	111.1
BaseCost_Time + 2	**A10**	4.8	11.2	23.7	12.3	49.7	122.5	7.4	37.9	96.9
BaseCost_Time - 1	**A11**	11.4	35.3	34.5	61.7	141.2	140.2	50.0	105.2	104.3
PrintCost_Time + 2	**A12**	5.6	23.0	22.1	19.5	85.6	139.9	13.5	61.7	116.2
PrintCost_Time - 1	**A13**	8.0	29.3	28.8	42.7	143.2	142.7	34.2	112.6	112.8
AssembleCost_Time + 2	**A14**	4.9	17.1	17.1	20.2	80.5	138.6	15.3	62.3	120.6
AssembleCost_Time - 1	**A15**	11.8	33.6	33.4	50.3	142.0	142.4	38.1	107.6	108.1
Print & Assemble Time + 2	**A16**	4.8	11.4	10.3	12.6	50.6	72.6	7.7	38.3	61.3
BaseCost_NonPr + 1	**A17**	7.8	23.0	23.3	34.5	90.1	90.8	26.5	66.2	66.6
PrintCost_NonPr + 1	**A18**	7.7	23.4	22.5	33.7	99.2	97.6	25.7	74.0	73.8
AssembleCost_NonPr + 1	**A19**	7.7	25.1	24.6	34.2	125.8	126.7	26.1	99.3	100.0
[All]CostPrintable + 1	**A20**	7.8	17.4	17.3	29.3	71.5	100.4	21.3	53.1	81.8
[All]CostPrintable + 2	**A21**	7.0	15.2	13.6	23.1	45.7	74.4	15.9	30.0	60.1
Print & Assemble Pr + 2	**A22**	7.4	18.0	17.8	28.5	74.3	100.6	20.9	55.4	82.0
Base & Print Pr + 2	**A23**	7.3	19.5	17.3	25.2	56.1	79.1	17.9	36.0	61.2
Base & Assemble Pr + 2	**A24**	7.5	15.3	15.9	28.2	60.6	94.4	20.6	44.6	77.8
[All]CostPrintable - 1	**A25**	8.0	36.9	35.9	35.3	138.6	136.6	27.1	100.2	99.4
AssembleCost_Pr - 1	**A26**	7.8	27.3	27.1	34.3	135.8	138.7	26.3	107.1	110.3
PrintCost_Pr - 1	**A27**	8.0	27.3	27.2	35.5	137.8	139.5	27.4	109.3	111.1
BaseCost_Pr - 1	**A28**	7.7	30.7	31.2	34.4	136.3	138.2	26.3	104.3	105.7
Print_Efficiency = 0.25	**B1**	7.3	14.6	14.0	28.8	41.8	52.7	21.4	26.9	37.8
Print_Efficiency = 0.5	**B2**	7.6	17.0	17.0	33.0	67.7	85.9	25.0	50.0	68.3
Print_Efficiency = 1.5	**B3**	8.0	29.4	30.0	34.9	137.8	137.6	26.8	107.2	106.4
Collect_Amount = 0.25	**B4**	7.8	19.3	16.2	34.1	59.7	102.2	26.1	39.9	85.3
Collect_Amount = 0.5	**B5**	7.9	21.4	20.6	34.2	91.2	131.0	26.2	69.1	109.3
Collect_Amount = 1.5	**B6**	7.9	26.0	26.8	34.9	137.7	142.3	26.8	110.2	114.4
Print_Amount = 0.25	**B7**	7.2	15.9	14.9	31.6	46.2	95.2	23.8	29.5	78.8
Print_Amount = 0.5	**B8**	7.7	19.2	19.7	34.6	91.3	142.0	26.7	70.7	120.7
Print_Amount = 1.5	**B9**	7.6	25.9	26.8	33.9	137.5	141.1	25.9	110.2	113.1
Collect & Print Amount = 0.5	**B10**	8.0	18.3	16.9	35.9	74.0	129.6	27.6	54.7	110.5
QualityThreshold + 0.1	**C1**	7.5	23.3	24.5	32.5	117.5	138.1	24.7	93.0	112.6
QualityThreshold + 0.2	**C2**	7.1	22.2	22.1	29.9	105.5	129.1	22.5	82.1	106.1
QualityThreshold + 0.3	**C3**	6.4	19.0	20.3	25.6	82.5	117.5	19.1	62.6	96.2
QualityThreshold + 0.4	**C4**	4.8	12.8	14.1	18.3	49.7	92.1	13.3	36.5	77.1
RiskAmount_Print = 1%	**C5**	7.7	24.6	24.5	34.4	124.3	139.9	25.3	92.0	108.0
RiskAmount_Assemble = 1%	**C6**	5.9	17.2	15.6	30.0	107.0	132.5	22.5	83.2	108.4

(continues)

Table B.9. Strategic-P decision-making algorithm results (continued).

Strategic-P	ID	Assembly Potential			Collection Potential			Print Potential		
		30	**50**	**70**	**30**	**50**	**70**	**30**	**50**	**70**
RiskAmount Pr & As = 10%	**C7**	1.0	2.0	1.3	11.9	18.8	21.2	3.6	4.5	5.5
RiskAmount Pr & As = 15%	**C8**	0.9	0.6	0.5	10.0	12.2	12.5	3.0	2.2	2.5
RiskAmount_Assemble = 15%	**C9**	0.6	0.6	0.3	9.8	14.7	12.7	5.5	7.8	7.8
Quality_incr_Chance = 0.01%	**C10**	7.5	23.9	24.6	34.0	124.3	138.8	26.2	98.7	112.8
Quality_decr_Chance = 25%	**C11**	7.7	25.5	25.6	34.8	131.2	144.2	26.9	104.6	117.5
Quality_decr_Chance = 75%	**C12**	7.7	23.7	23.2	33.7	116.3	132.9	25.7	90.9	108.1
Quality_decr_Upper = 0.5	**C13**	7.3	20.2	20.3	30.0	98.6	123.5	22.6	76.5	101.1
Qual_incr Chance & Upper * 2	**C14**	7.9	24.6	24.0	34.6	125.7	138.7	26.6	99.6	112.8
RiskQuality_Modifier = 10.0	**C15**	7.9	23.1	22.8	33.9	117.9	136.6	25.8	93.5	111.7
RiskQuality_Modifier = 25.0	**C16**	7.5	19.0	19.9	31.8	92.1	125.9	23.9	71.1	103.9
RiskFactory_Modifier = 0.5	**C17**	7.9	24.6	24.4	34.6	126.5	140.0	26.4	100.6	114.3
RiskFactory_Modifier = 1.0	**C18**	7.9	24.7	24.6	34.0	127.4	139.5	25.9	100.7	113.4
Quality Thres & Chance	**C19**	3.8	9.2	7.0	11.8	30.4	39.4	8.0	20.9	31.9
Initial_Printable / 2.0	**D1**	7.6	21.0	20.4	31.2	99.9	142.5	23.5	78.4	120.8
Initial_Printable * 2.0	**D2**	7.8	35.2	34.1	34.4	140.5	138.6	26.4	104.1	102.9
Initial_Materials = 0	**D3**	7.6	22.6	22.2	34.0	109.3	134.8	26.1	85.4	111.1
Initial_Materials / 2.0	**D4**	7.3	23.7	24.2	32.8	118.2	141.8	25.1	92.4	116.4
Initial_Materials * 2.0	**D5**	7.9	26.7	26.8	35.6	136.6	141.1	27.6	108.1	113.3
Env_Materials / 2.0	**D6**	7.9	24.7	25.3	34.8	94.9	95.8	26.8	69.3	70.0
Env_Materials * 2.0	**D7**	7.8	24.3	22.0	34.7	124.4	128.7	26.5	98.4	105.1
Env_Materials * 100	**D8**	7.9	25.1	19.5	34.9	125.9	119.5	26.8	99.6	97.5
Initial_NonPr / 2.0	**D9**	7.9	22.9	22.4	34.6	68.8	66.9	26.5	45.0	43.7
Initial_NonPr * 2.0	**D10**	7.7	24.3	23.6	33.7	123.7	151.4	25.6	98.0	125.5
Initial NonPr & Env * 2.0	**D11**	7.9	25.4	22.9	35.0	125.5	225.6	27.0	98.5	199.1
Initial NonPr & Env * 2.0, Raw=0	**D12**	7.9	23.2	22.6	34.5	110.7	214.8	26.4	86.5	190.4