Use of Genetic Algorithms and Transient Models for Life Support Systems Analysis

Luis F. Rodríguez [*]

University of Illinois at Urbana-Champaign, Urbana, IL 61801

Scott Bell [†]

NASA Johnson Space Center/SKT Inc., Houston, Texas, 77058, USA

David Kortenkamp [‡]

NASA Johnson Space Center/Metrica Inc., Houston, Texas, 77058, USA

Direct experimentation with physical systems is slow, expensive, and must wait until the physical system is built. Simulations allow for the testing of different system configurations before hardware is built. This is useful where malfunctions are reliably expected, costs are high, or where the integrated systems are unavailable. Thus, it is important that such simulations are transient and integrate all major subsystems and activities. This paper describes a habitat model that is transient, discrete, stochastic, and non-stationary. It models most of the components of a life support system including the crew, crops, water and air recovery systems, EVAs and power. Since the model accepts non-stationary input, it can be used to test habitat configurations and components before building an actual habitat. Malfunctions can be injected at any time. A genetic algorithm is utilized here to find an optimal habitat configuration for a ninety day mission to the Moon. Approximately 20,000 system configurations were considered in a series of experiments considering several Lunar scenarios. It has been shown that a tuned genetic algorithm is capable of preliminary sizing several life support components. In addition, the fitness function serves as proxy for equivalent system mass, a metric related to launch costs.

[*]Assistant Professor, University of Illinois at Urbana-Champaign, Department of Agricultural and Biological Engineering, 376B Agricultural Engineering Sciences Building, MC-644, 1304 W. Pennsylvania Avenue, Urbana, IL 618101

[†]Scientist, SKT Inc., NASA JSC Mail Stop ER2, Houston TX 77058

[‡]Senior Scientist, Metrica Inc, NASA JSC Mail Stop ER2, Houston TX 77058

Nomenclature

a_i	Value of attribute i
$a_{i,max}$	Maximum value of attribute i
f	Fitness
hr	Time, hours
kg	Mass, kilograms
mol	Moles of compound
m^2	Area, squared meters
t	Simulation time, hours
w_i	Weight of attribute i
w_t	Weight of simulation time, hrs^{-1}
K	Temperature, K
L	Volume, liters
W	Power, watts

Introduction

NASA has embarked on a new exploration strategy that will return people to the Moon and eventually to Mars.[1] A key component of a successful exploration strategy is life support systems, which drive launch mass and mission cost. And a key aspect of life support systems is sizing and optimization of the interconnected life support components. Technology choices, buffer sizes, power plant sizing, crop planting area and subsystem flow rates all need to be determined in order to design an appropriate and optimal system that meets mission requirements. Often these decisions are made using steady state simulations and labor intensive searches .[2] Our hypothesis is that dynamic, transient models and automated search tools, such as genetic algorithms, can assist in the design of habitat life support systems.

To explore our hypothesis we: 1) created a generic, dynamic simulation of a habitat life support system; 2) represented the design decisions of the habitat in a genetic algorithm; 3) developed a fitness function to judge habitat designs and; 4) ran the genetic algorithm for tens of thousands of simulations to search for an optimal life support design. The results show that automated search tools can play an interesting and under-appreciated role in spacecraft design.

Related Work

Dynamic simulations have been prevalent in many fields for years, but they are just now becoming common in life support analysis.[3] Most previous dynamic simulations have been specific to existing or planned hardware implementations. For example, the BioPlex dynamic simulation[4] and the MELiSSA simulation.[5] A notable exception is work by Luis Rodriguez on top-level modeling of advanced life support simulations.[6] Some dynamic simulations are developed to aid control and diagnosis of life support systems instead of design.[7] Our modeling approach is unique in that it is a general habitat simulation and can be easily configured for a variety of different situations. It is suitable for both analysis, as described in this paper, and control applications.[8]

Genetic algorithms were invented by John Holland to allow computers to mimic natural selection.[9] Since then they have a long history of being used for optimization and design problems. These applications range from wing design[10,11] to concert hall acoustics.[12] There have been genetic algorithm applications for designing space structures[13] and space telecommunications.[14] Biology has applied genetic algorithms to molecular design.[15] Computer scientists have applied genetic algorithms to network routing,[16,17] to programming[18] and to job shop scheduling.[19] Financial applications include stock forecasting[20] and exchange rate forecasting.[21] Materials scientists have long used genetic algorithms in their design work.[22,23] Other applications range from earthquake epicenter location[24] to wind turbine optimization.[25] Interesting applications of genetic algorithms on robotics have led to the design of fantastic creatures optimized for particular applications[26] and to novel control techniques.[27] Finally, there has been other work on controlling advanced life support systems that complements our work on designing advanced life support systems.[28]

Overview

The next section of this paper describes our dynamic model of a life support system and its application to our analysis domain. We then present our genetic algorithm, including our representation of a gene, our fitness function and our selection approach. Finally, we describe a series of experiments that used the genetic algorithm and the dynamic model to search a space of possible habitat configurations and choose an optimal configuration. We analyze the results of the experiments and discuss future work.

Modeling a Lunar Habitat

This habitat simulation is based upon a previously developed model known as BioSim.[29] It is a generic habitat simulation, so the first task was to create a specific instance of BioSim for a Lunar habitat. Instances of BioSim are specified in an eXtensible Markup Language (XML) file, thus creating a new instance of BioSim does not require changing any computer code. The XML file is read by the BioSim simulation at start-up and the appropriate configuration is instantiated. This makes creating different habitats very easy even for those with no programming experience.

Simulation Overview

A typical habitat system consists of multiple interacting modules.[30] The models are top-level process models that take in certain resources and produce other resources. They are not, however, component models. That is, they do not model physical objects such as valves, pumps, etc. Figure 1 shows the modules in our simulation. Detailed descriptions of each follow.

Crew

The crew module is implemented using models described by Goudarzi and Ting.[31] The number, gender, age and weight of the crew are settable as input parameters. The crew cycles through a set of activities (sleep, maintenance, recreation, exercise, etc.). As they do so they consume oxygen, food and water and produce carbon dioxide, dirty water and solid waste. The amount of resources consumed and produced varies according to crew member attributes and their activities. The crew's activities can be adjusted by passing a new crew schedule to the crew module. A default schedule can also be used. The crew module is connected to a crew environment that contains an atmosphere that they breathe. The initial size and gas composition of the atmosphere are input parameters. As the simulation progresses the composition of gases in the atmosphere changes.

Biomass

The biomass module models crops that produce food and oxygen during the simulation while consuming carbon dioxide and water. Currently, nine crops are modeled: wheat, rice, soybean, tomato, lettuce, dry bean, peanut, sweet potato, and white potato. The growing area and crop mix of the biomass module are selected by the user before simulation is executed. Within this growing area the amount of the different crops can be adjusted during a simulation either through the crew interface or directly through the biomass interface. As crops grow they consume carbon dioxide, potable or gray water and require photosyntheticly active light. At the same time, they produce oxygen and transpire water. The crops also produce biomass when they are harvested. The production and consumption of resources are modeled according to Barta, et al.,[32] Jones and Cavazzoni,[33] and the Baseline Values and Assumptions Document (BVAD) published by NASA's Advanced Life Support (ALS) Program.[34] The biomass module has its own environment that contains an atmosphere for the plants. The default configuration has separate biomass and crew environments because the ideal atmospheric composition for growing plants is not necessarily ideal for humans. The model, however, is reconfigurable. Alternatives include initialization with a single environment for crew and crops or with multiple environments for different crops. Harvesting and planting of crops is currently done automatically. It is assumed that the same crop is sown again once it has been harvested. However, a different crop schedule can be passed to the biomass module. Artificial lighting and water is provided to crops; the amount of lighting and water available to the crops is adjustable.

Food Processing

The food processing module takes in harvested crops and produces edible and inedible biomass. The inedible biomass is further divided into inedible dry mass, which goes to the solid waste store and inedible water mass, which goes to the dirty water store, simulating a drying process. Edible biomass is converted into a food object type that has three properties: calories, water, and mass. Each of the nine crops are converted into nine different food types. The food processing module adds these food types to the food store, which may contain other user defined food not produced by the biomass production system. When the crew module needs food it asks the food store for a specific number of calories. The food store creates a mix of food types to provide those calories and passes them to the crew module, while deleting that food from the food store. The crew module uses the water content of food to reduce the amount of potable water consumed by the crew. In the future, adding the ability to request specific menus and to balance other nutritional requirements in addition to calories would be highly desirable.

Air Revitalization

The air revitalization module consists of several subsystems that provide breathable air to the crew environment. The Variable Configuration Carbon Dioxide Removal System (VCCR) takes air from the crew atmosphere, removes carbon dioxide, and returns the remaining air mixture. The removed carbon dioxide is collected in a dedicated storage. The Carbon Dioxide Reduction System (CRS) takes carbon dioxide from the store and reacts it with hydrogen, making water and methane. The methane is vented and the water is passed to the third system, the Oxygen Generation System (OGS). The OGS electrolyzes the water into hydrogen and oxygen. The oxygen is stored and the hydrogen is returned to the CRS. At any time, the air revitalization module may take water from the potable water store for use with the OGS, or it may return excess water to the buffer. In addition, an accumulator extracts oxygen from the biomass atmosphere and places it in storage. Injectors are available to take gases from the stores and inject them into the atmospheres. A control challenge is to maintain an optimal gas mixture in the crew and biomass environments while minimizing both energy use and storage sizes. The capacities of the stores can be assigned at initialization. All modules require power to function, although the OGS requires substantially more power than the other subsystems.

Water Recovery

The water recovery module handles high strength wastewater and gray water with the goal of producing potable water. The water recovery module consists of four subsystems that process the water. The biological water processing (BWP) subsystem removes organic compounds. Then the water passes to a reverse osmosis (RO) subsystem, which further processes the stream, but loses 15% of the water to the RO brine stream. The RO brine is passed to the air evaporation subsystem (AES), which recovers the remainder through a drying process. These two streams of water (from the RO and the AES) are passed through a post-processing subsystem (PPS) to be polished to potable water standards. An external controller can turn on or off various subsystems. For example, all water can pass through the AES at a higher energy cost. The design of this wastewater processing system is based upon a series of integrated life support system tests executed from 1996–2000 administered by Edeen and Barta,[35] Brasseaux, et al.,[36] and Edeen.[37]

Extravehicular Activities

Extravehicular activities (EVA) involve crew members leaving the habitat for variable periods of time. While outside the habitat, the crew members do not place a metabolic load on the habitat life support systems, but rather utilize alternate systems. We implement this in our simulation by creating a miniature habitat with parameters set by the user. This mini-habitat exists for the duration of the simulation. The mini-habitat has its own environment, stores, recycling systems, etc. As an example, the mini-habitat could be an extravehicular mobility unit (EMU) with small air volume, water stores, air revitalization system, and power supply. This mini-habitat would be suitable for one crew member. While the crewmember is inside the mini-habitat, the crew member would continue natural metabolic operations, such as respiration and excretion, however not drawing from the original environment. When the EVA ends, the crew member returns to the main habitat and resumes consumption of resources. Because all parameters of this mini-habitat are settable, it could represent any EVA hardware, from space suits to pressurized vehicles capable of holding multiple crew members for extended periods. For set-up and control purposes the mini-habitat is no different from any other habitat simulation and uses the same underlying models. EVAs can be scheduled dynamically at any time and for any duration. Multiple mini-habitats can exist at the same time.

Solid Waste

The current solid waste module models an incineration process in order to recover carbon. The module takes dry waste from the crew and food processing modules and oxygen to produce carbon dioxide. We plan to add a lyophilization system to the solid waste module in the future.

Power

The power production module supplies electricity to all of the other modules. There are two models for this module. One simulates a nuclear-style power system that supplies a continuous, fixed amount of power. A second simulates a solar-style power system that supplies a varying amount of power. An external control program can set the amount of power going to each module up to the total amount of power available.

Habitat Parameters

The previous section, entitled *Simulation Overview*, describes the default habitat simulation. For the experiments described here, a specific instance of the simulation reflecting a Lunar habitat was designed based upon a draft reference mission.[38]

The Lunar reference mission assumes a four person crew with equal numbers of men and women. A mission is 90 days with the habitat initiated and operating nominally upon crew arrival. The landing site is the Lunar south pole with the sun above the horizon 80% of the time and surface temperatures between $210K$ and $230K$ during the day. The habitat atmosphere is composed of 29% oxygen at an overall pressure of 65.5 kPa and a leakage rate of 0.00224 kg/day. Food is shipped in most circumstances (although we looked at the addition of small salad crops) and is 0.257 kg/crewmember-day moist food and 0.665 kg/crewmember-day of dry food. Air, water, and waste recovery systems are part of the habitat.

Parameters such as the atmospheric volume of the habitat, the size of the recovery systems, the amount of salad crops and the size of the power subsystem were not fixed and were determined through analysis of simulation results as described in *Analysis* Section.

Extravehicular Activities

As described in the *Extravehicular Activities* Section, EVAs can be scheduled at any time. Following the Lunar reference mission our simulation sends one crew member on a four hour EVA each day of the mission.[38] An EVA takes place through an airlock that is 3.7 m^3 in size and 10% of the airlock atmosphere is lost each time the airlock is used.

Malfunctions

Our simulation environment allows the user to insert malfunctions into any subsystem, at any time. Malfunctions can also be insert randomly by the system. These malfunctions can be permanent or temporary. For all of the experiments described in this paper two malfunctions were introduced into the habitat. First, there is an oxygen storage malfunction at 200 hours into the mission. This results in both a loss of 50% of the storage capacity and 50% of the oxygen in the store at that time. The capacity reduction is permanent. Second, there is a power production malfunction at 502 hours into the mission that results in a permanent loss of 50% of the power production capability.

Lunar Habitat Scenarios

The goals of the experiments described in this paper were to determine appropriate system-sizing requirements for the habitat life support systems and to compare different habitat configurations. We examined five habitat configuration variations based on the previously described habitat parameters:

1. No biomass production facility is included; all food is presupplied for the mission; all air and water recovery is performed by physico-chemical subsystems.
2. A small biomass production facility is fully integrated with the crew cabin air and water recovery subsystems.
3. A small biomass production facility is integrated with the crew cabin air recovery subsystem, but has an independent water recovery subsystem.
4. A small biomass production facility is integrated with the crew water recovery subsystem, but has an independent air recovery subsystem and atmospheric environment.
5. A small biomass production facility with independent water and air recovery subsystems and atmospheric environment.

The biomass production facility of these experiments includes only lettuce and tomatoes.

Interfacing with the Simulation

BioSim has an Application Programmers Interface (API) that allows external programs to access BioSim functionality. The API is written using the Common Object Request Broker Architecture (CORBA), which standardizes network access.[39] Thus, a program can be running on one computer and interfacing with BioSim running on another computer. The use of CORBA also means that external programs can be written in any language that supports CORBA, including C, C++, Java, LISP and many others. The API can be used to configure BioSim, to control any of its modules, to inject failures and to step BioSim through discrete time intervals. The API can also be used to read data from BioSim, including module status, store levels, crew activities, among many others.

In the experiments described in this paper, an external genetic algorithm (GA) searches for an optimal habitat configuration. The GA provides us with the ability to rationally search a wide array of configurations for optimal solutions. Thus, in this case the GA is capable of instantiating, configuring, executing, and evaluating BioSim simulations. Details describing the GA are provided in the following section.

Genetic Algorithms

A genetic algorithm[9] encodes model inputs as a 'gene,' often in the form of a binary string (although we use a set of floating point values in these experiments). The models are simulated and a fitness function is utilized to evaluate the quality of the gene. The process starts with a randomly selected population of genes. Each is simulated and evaluated. The best performing genes are saved while the worst genes are eliminated. The best genes are then *mutated*, *inverted*, or *crossed* creating a new population, which are direct decedents. Thus, to complete the evolutionary analogy, beneficial traits are preserved through subsequent generations, because only the best performing genes will be allowed to reproduce.

Some implementations of genetic algorithms use bit strings to represent genes. Instead of a bit string, we used float values for each attribute in a gene. An attribute can be anything, from the capacity of a water store to the size of the crew cabin. To derive numbers in the range that we want, we *mod* the float by the maximum value the attribute can have—the *mod* operator divides the value by the maximum value and takes the remainder, which will always be between 0 and the maximum value. For example, a tomato shelf can have a maximum size of 20 meters squared. To derive a value from the underlying float, we *mod* the number by 20 giving us a value between 0 and 20.

Performance is defined by a fitness function specified by the user. In our analysis, only the top third of genes were utilized for the generation of the subsequent populations. Details describing the fitness function utilized here are included in the *Fitness Function* Section.

For crossover operations, half of a gene's attributes are combined with half of another gene's attributes. For mutation operations, five attributes in a gene were changed to random values. For inversion operations, all the float values representing attributes were interchanged. For example, the last attribute would become the first, the previous first attribute becomes second and so forth. In each case, attributes are scaled appropriately as described by the mod function above. In this implementation, crossovers, mutations, and inversions respectively occur 10%, 80%, and 10% of the time. Initial attribute values for each gene were randomly assigned. By implementing the genetic algorithm with a large proportion of random mutations, it is expected that the search space will be evenly considered, reducing the likelihood of finding local minima or maxima.

Genes

A gene in our genetic algorithm represents a life support system configuration. Depending upon the scenario (see Lunar Habitat Scenarios Section) a gene will have between 12 and 20 attributes, each with an integer value. At first these integer values are randomly assigned to each attribute. As each generation is simulated, the GA adjusts each attribute's value in search of an optimal solution. The attributes and the range of values they can take are listed in Table 1.

Fitness Function

The intention of this fitness function is to identify optimal designs for the 90 day Lunar mission described in the *Habitat Parameters* Section.An optimal design will consume minimal resources while meeting mission objectives. The fitness function utilized is:

$$f = w_t t + \frac{\sum_{\forall i}(w_i a_{i,max} - w_i a_i)}{\sum_{\forall i}(w_i a_{i,max})}$$

where f is the fitness of a configuration, t is the length of the mission in hours, w_i is the weight of attribute i, a_i is the value of attribute i and $a_{i,max}$ is the maximum value of attribute i (the list of attributes can be found in the *Genes* Section. The function f is a unit-less measure of fitness. Thus, the weight w_t has unit 1/hr and takes a value equal to one. The first term, $w_t t$, measures the number of simulated hours without system failure in integer values, as BioSim is a discrete event simulation. The second term will never be greater than one, which allows the first term to dominate. That is, a mission lasting 2160 hours will always be rated as more fit than any mission lasting 2159 hours or less, regardless of the amount of resources consumed. In this analysis, we are most concerned with 90 day Lunar missions (2160 hours), so we have instructed BioSim to terminate any simulations that complete 90 days. This renders the first term equal to 2160 for all 90-day missions, whether or not a configuration may have been capable of surviving longer than that time. Since all 90-day missions will be more fit than any mission lasting less than 90 days, the second term in the fitness function differentiates configurations, selecting those that minimize resources. The second term considers the sizing of the various subsystem attributes, their relative weights, and rewards configurations with components sized smaller than their maximum values. The attributes utilized and their weights are shown in Table 2.

The weights utilized in this analysis have been roughly derived based on ESM, and as such allow comparison of system resources on a mass basis. When sizing components within the system the GA will vary the attributes listed in the *Genes* Section.For example, the power production system is sized based upon the maximum power output necessary to sustain the system. In this analysis, this suggests that the value chosen will be the power output at Lunar noon. This aspect of the system is weighted using the cost equivalency specified in the BVAD, 0.062 kg/W.[34] Power storage, crew and crop volumes, and food storage are similarly weighted using cost equivalencies.[34]

Specific cost equivalencies are not available for the remaining attribute weights so these have been derived on a case specific basis. For this purpose, it is critical to understand the manner that BioSim simulations handle power distribution during a simulation. During each hour increment within the simulation, the power system distributes power to the various systems that are scheduled to work either directly from the power production system or power storage. If at this time the available power is not enough to power all systems, some will not operate. The order in which components within the system are allocated power is randomly selected each tick. Thus, within BioSim there is an impetus to size components to fit within the available power supplied by the power systems. If not, components will fail to operate due to lack of power which may lead to a system failure.

Similarly, as components are sized they must fit within the envelope defined by the crew and crop volume attributes. However, currently there is no oversight procedure within BioSim to ensure that components sized by the GA fit within the volume, leaving enough living space for the crew. This will be addressed in future versions of BioSim and at this time it is assumed that component sizing is not limited by volume. Nevertheless, the volume does define the available oxygen and carbon dioxide in the atmosphere and this might affect the performance of some unit processes within the system.

Based on the above power and volume conventions within BioSim, the remaining attribute weights are generally sized strictly on a mass basis. The assumption is that both power and volume issues are addressed by BioSim. Cooling issues are generally proportional to power issues and are currently not incorporated in the fitness function, although it is anticipated that this will change in the future. Further, crew time issues are not currently addressed, but will be if sufficient data becomes available.

With the exception of the oxygen injector, the data source utilized for component mass data was the ALSSAT Tool utilized to determine the Advanced Life Support Metric for the Advanced Life Support Project within NASA. Data for this analysis was provided through personal communication with the principal developer of the tool.[40–42] In each case, specific masses were derived based on the key parameter selected by the GA. For example, a salad machine was sized in the 2004 determination of the ALS Metric.[40] This salad machine had a mass of 555.15 kg with an area of 7.5 m^2. Our assumption is that half of this area will be devoted to both lettuce and tomato crops. Thus, the value of the weight for crop shelf sizes, based on a specific area, is 37.010 $\frac{kg}{m^2}$. This has been done similarly for the oxygen injector and the storage of oxygen and all forms of water. As data was not readily available for the mass of the oxygen injector, an assumption was made based upon communication with other analysts in ALS. This weight is relatively small, thus it is not expected that it affects the results appreciably. When converting these mass values to specific weights arbitrarily large amounts of air (1×10^6 $mols$) and water (1,000 L) were assumed for the model systems described in the metric analyses.

The only remaining attribute weights are the VCCR, OGS, and WRS. These systems are sized based on how much power is consumed to process their daily design loads. At the time of this analysis, a satisfactory weighting relation for power consumption to processor mass had not been identified, so a zero weight has been utilized. The implicit assumption is that the actual mass of these processors is negligible relative to their power, volume, and cooling requirements.

Analysis

We used the genetic algorithm described in the previous section on the five scenarios described in *Lunar Habitat Scenarios* Section. This section explains our experimental methodology and our preliminary results.

Experimental Setup

Approximately 20,000 simulations were executed using the model and the genetic algorithm. The experimental setup included a genetic algorithm handler program that managed the genes and the fitness function. The handler had access to fourteen instances of BioSim running on eight different computers. The first step in the experimental process was for the handler to randomly create as many genes as there were instances of BioSim; for the purposes of this explanation, assume there are twelve instances of BioSim. The handler then sends each instance of BioSim one of the twelve randomly created genes. Each instance of BioSim executes its gene, meaning it configures the BioSim model according to the gene (see the *Genes* Section),and then runs BioSim until either the mission ends due to lack of resources or the baseline 90 day mission length is reached. It then sends its fitness back to the handler. The handler keeps genes that score in the top third, or four genes in our example. Then, amongst these four genes we cross, mutate, and inverted them to create the next generation of genes. Crossing happens 10% of the time and occurs by taking half of one gene and half of the other. Mutations happen 80% of the time and occurs by modifying one random parameter in the gene. Inversion happens 10% of the time and occurs by are shuffling the parameters within the same gene. All genes are then sent back to a BioSim instance for simulation and evaluation. The process happens asynchronously as simulations may run for a long or a short period of time. Thus, crosses, mutations, and inversions happen constantly and simulations happen constantly with the top third of the genes being saved. This process continues until approximately 1000 simulations have been executed, which we term trials. After 1000 trials we start the process over again with a new set of randomly chosen genes. This is replicated four times for each of the five configurations described in the

Lunar Habitat Scenarios Section.Thus, the preliminary results described in the next Section are based on 20,000 simulation runs. Configurations are numbered sequentially (1-5) as they are listed in the *Lunar Habitat Scenarios* Section.The four replications of each experiment are numbered alphabetically (A-D).

Preliminary Results

Several interesting results have been observed. Some level of constraint verification and modified weights needs to be developed for BioSim and the GA, thus results in this section should be considered preliminary. Plans have been prepared to complete this analysis in the future. At this time, several configurations that survive the desired 90-day mission have been identified. Several components within the system have been sized to some extent, although some reduction of the error bounds would be desirable.

Figure 2 demonstrates what a prototypical GA output may look like. Charts such as Figure 2 assist in determining if the GA approached a useful solution. Diamonds mark the fitness of individual trials of BioSim representing the performance of a gene chosen by the GA. Along the horizontal axis are the number of BioSim trials executed by the GA. For our experiments it is typical to have on the order of 1000 trials. The best fitness observed thus far is charted by the thin line. It can be observed in Figure 2 that the fittest configuration increases in quality as the GA completes more trials. Eventually the GA reaches the desired 90 day (2160 hr) threshold. At this scale, it is not possible to observe further increases in the fitness function, although they do occur, as the thin line is hidden behind the diamond markers.

The thick line represents the average fitness observed during trials up until that trial. As the fittest configuration increases in quality, it can be observed that the average fitness also increases. This indicates that the GA is in fact performing as desired, increasing the fitness of configurations it chooses, thereby increasing the average fitness.

Figure 3 shows an example of a parameter that was not conclusively sized by the GA, although this configuration did approach the desired 90-day mission. In this figure, the diamonds indicate the sizing chosen by the GA in a simulation as related to the resultant fitness. The failure of the GA to size the VCCR is most apparent at the extreme right of the figure where a distinct column of diamonds appear, indicating that multiple solutions of various sizes all have nearly equal fitness. The fitness at the extreme right of the chart corresponds to that of 2160, or a 90 day mission. This indicates that precise VCCR sizing is not critical for successful execution of a 90 day mission in this configuration.

Alternative output is demonstrated in Figure 4, where food storage is related to system fitness. In this case, it is apparent that mission viability is closely related to the available food in the system. For example, no amount of food less than approximately 600 kg was able to sustain a crew beyond 1500 hours. It is clearly shown that the minimum amount of food required for a viable mission is proportional to the mission length. Excess food, however, is not a limiting factor.

Similarly, maxima can be identified by the GA, as shown in Figure 5. As a general theme, power consumption has been identified by the GA as a critical issue. Maximum power consumption values, such as that pictured in Figure 5, have been identified in each configuration for many components. One notable exception is the VCCR component in configuration 1 (see Figure 3).

In some cases, minima and maxima can be identified for an attribute varied by the GA. Figure 6 shows that a minimum amount of oxygen generation is required for this configuration to sustain a proper oxygen partial pressure, however excessive oxygen generation overtaxes the power system.

It has been observed that it can be worthwhile to consider the range of the fitness function that minimizes the consumption of resources. For example, Figure 7 demonstrates that reductions in power storage were capable of increasing fitness when comparing systems that achieve the 90-day goal. This is likely due to the relative weight of power storage.

Interestingly, the VCCR can be sized very precisely in configurations 2-5. Figure 8 shows that systems that survive a complete 90 Day mission must have a VCCR sized to consume slightly less than 3 kW.

By considering the 4 replications of each GA analysis of each configuration it is possible to preliminarily size several components within the system. Referring to Figure 9 we observe that some components are sized more precisely than others, indicated by tight error bounds. It is theorized that tighter error bounds suggest a component has a more profound effect upon fitness. Further, Figure 10 demonstrates how the VCCR was not precisely sized in configuration 1, while it was sized quite precisely in configurations 2–5. It also has been theorized that since configurations 2–5 all include a crop subsystem, this may have prevented the GA from over sizing the VCCR, which would have removed the carbon dioxide necessary for crop growth.

The term $\sum_{\forall i}(w_i a_i)$ from the numerator of the second term of the fitness function can be utilized as a proxy for ESM, provided that the weights are correctly representative of the metric. As is the case for ESM, small values for the proxy are also indicative of preferred configurations. Averaged values from the replicates of each GA run for each configuration are displayed in Figure 11. As expected, there is a distinct cost for including crops within the system. Interestingly, the inclusion of independent water and air revitalization subsystems for the crew and crop subsystems greatly increases the value of the ESM proxy by over an order of magnitude. It is not likely that such a configuration will be preferred from an ESM perspective. It is theorized that this increase is due to the lack of carbon dioxide additions to the crop environment, which would otherwise be provided by crew respiration within an integrated atmosphere. This causes the GA to select a large and very costly crop environment volume to provide the necessary carbon dioxide and buffering capacity.

Conclusions and Future Work

This paper demonstrates the use of dynamic simulations and automated search tools in configuring and sizing Lunar habitats. Dynamic simulations allow for inserting run-time failures into habitat systems and taking those failures into account during the design process. Automated search tools can explore vast search spaces, identifying specific configurations that produce optimal designs. In this case, we searched through 20,000 configurations within a search space consisting of more than 2^{24} possible combinations. The primary conclusion of this preliminary study is that dynamic simulations and automated search tools can help a human designers or design teams identify optimal configurations and search for interesting combinations that might never have been considered. The ability to repeatedly and rapidly simulate habitat configurations and to change those configurations in mid-stream allows a designer much greater flexibility. The simulation, the automated search tool, and the fitness function can all be replaced with different algorithms and only one possible combination was demonstrated here.

In addition, several key new research paths have been identified. Firstly, in the event of the development of an integrated test bed for life support technology, validation of models results will be a critical step. In the meantime, however, verification of results can be performed via comparison with other generally accepted modeling tools. Anticipating limited access to an integrated test bed a consortium of the modeling community considering life support systems analysis should dedicate themselves to verifying model results across tools, platforms, perspectives, and techniques. Only through such an organized effort can confidence in model results be increased to a level that will truly bolster the concurrent technology development activity. A genetic algorithm, or other heuristic technique, might be a suitable tool for the standardization of models. Models and techniques might be tested by solving for an optimal scenario with a pre-specified configuration designed for benchmarking or standardization. The genetic algorithm could consider the possible scenario search space by specifying this through its genes. Reasonably analogous results with multiple models would give the community confidence that a given set of modeling tools is sound. The community could then consider and build upon additional analyses with these rated tools with increased confidence.

Secondly, it has been observed that the output of genetic algorithm seems to classify itself into tiered categories of performance based on our fitness function. It has been theorized that certain patterns may exist within the genes selected by the genetic algorithm that are inherently more successful than others. It is proposed that a pattern recognition analysis should be performed upon the results of these analyses. If inherently successful patterns can be identified, then it is expected that the sensitivity of those results may also be determined. Three benefits might arise: (1) inherently successful designs can be targeted for later analysis and implementation; (2) inherently flawed designs can be studied to identify their detrimental traits and (3) genetic algorithm fitness functions and crossing, mutation, and inversion rates can be modified to improve analysis performance.

Finally, future analyses of life support systems utilizing genetic algorithms (or other heuristic search techniques), especially those for long term exploration missions, are suited for the consideration of unconventional system configurations and scenarios with multiple objectives. Analysts should take advantage of the ability of the GA to search a wide array of configurations and scenarios. Consider the problem of providing healthy diets for the crew in a long-term exploration mission. Constraints include physical and mental health of the crew and shelf life, in addition to the customary mass, storage, energy and crew time limitations. Food can be either pre-supplied rations, or grown and processed in-situ. The design of a system including a mixture of food sources is a multi objective problem which might have an unconventional solution when applied to a long-term life support system. Understandably, this system is starkly different from other systems previously designed and delivered by NASA due to the long mission duration and the consideration of biological unit processes. Unconventional solutions should be anticipated for such problems. Utilizing a GA for the analysis of this type of problem will provide confidence in such solutions because a wide array of potential solutions have been tested via the model. Further, solutions will be supported by quantitative valuations from the fitness function as well as any figures of merit derived, such as the proxy for equivalent system mass described here. Promising solutions will still need verification and validation elsewhere, but this will be required of results derived from conventional techniques as well. Thus, the use of heuristic design tools, like genetic algorithms, especially early in the development life cycle can potentially provide significant savings in research, design, and mission costs and merit serious consideration by the life support research community, much like their conventional analysis counterparts.

References

[1]National Aeronautics and Space Administration, “Exploration Systems Interim Strategy,” Tech. Rep. NP-2004-07-362-HQ, NASA Headquarters, 2004.

[2]Yeh, H., Brown, C., Jeng, F., Lin, C., and Ewert, M., "ALSSAT Development Status and its Applications in Trade Studies," *Proceedings of the 34th International Conference on Environmental Systems*, 2004.

[3]Jones, H. W., "Nonlinear Transient Models in Life Support," *Proceedings of the 32nd International Conference on Environmental Systems*, 2002.

[4]Finn, C. K., "Dynamic System Modeling of Regenerative Life Support Systems," *Proceedings of the 29th International Conference on Environmental Systems, SAE paper 1999-01-2040*, 1998.

[5]Ordonez, L., Poughon, L., and Waters, G., "MELiSSA Higher Plants Compartment Modeling Using EcosimPro," *Proceedings of the 34th International Conference on Environmental Systems, SAE paper 2004-01-2251*, 2004.

[6]Rodriguez, L. F., Kang, S., and Ting, K., "Top level modeling of an ALS System utilizing object oriented techniques," *Advnaces in Space Research*, Vol. 31, No. 7, 2003.

[7]Biswas, G., Manders, E. J., Ramirez, J., Mahadevan, N., and Ablewahed, S., "Online Model-Based Diagnosis to Support Autonomous Operation of an Advanced Life Support System," *Habitation: International Journal for Human Support Research*, Vol. 10, No. 1, 2004.

[8]Klein, T., Subramanian, D., Kortenkamp, D., and Bell, S., "Using Reinforcement Learning to Control Life Support Systems," *Proceedings International Conference on Environmental Systems*, 2004.

[9]Holland, J., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor MI, 1975.

[10]Obayashi, S., Sasaki, D., Takeguchi, Y., and Hirose, N., "Multiobjective evolutionary computation for supersonic wing-shape optimization," *IEEE Transactions on Evolutionary Computation*, Vol. 4, No. 2, 2000.

[11]Sasaki, D., Morikawa, M., Obayashi, S., and Nakahashi, K., "Aerodynamic shape optimization of supersonic wings by adaptive range multiobjective genetic algorithms," *Lecture Notes in Computer Science*, edited by K. Deb, L. Theile, C. Coello, D. Corne, and E. Zitler, Springer Verlag, Berlin, 2001.

[12]Sato, S., Otori, K., Takizawa, A., Sakai, H., Ando, Y., and Kawamura, H., "Applying genetic algorithms to the optimum design of a concert hall," *Journal of Sound and Vibration*, Vol. 258, No. 3, 2002.

[13]Keane, A. and Brown, S., "The design of a satellite boom with enhanced vibration performance using genetic algorithm techniques," *Proceedings of the Second International Conference on Adaptive Computing in Engineering Design and Control*, 1996.

[14]Williams, E., Crossley, W., and Lang, T., “Average and maximum revisit time trade studies for satellite constellations using a multiobjective genetic algorithm,” *Journal of the Astronautical Sciences*, Vol. 49, No. 3, 2001.

[15]Glen, R. C. and Payne, A. W. R., “A genetic algorithm for the automated generation of molecules within constraints,” *Journal of Computer-Aided Molecular Design*, Vol. 9, No. 1, 1995.

[16]Xu, Z. and Wu, A. S., “Adhoc-like routing in wired networks with genetic algorithms,” *Ad Hoc Networks Journal*, Vol. 2, No. 1, 2004.

[17]He, L. and Mort, N., “Hybrid genetic algorithms for telecommunications network back-up routeing,” *BT Technology Journal*, Vol. 18, No. 4, 2000.

[18]Koza, J., Keane, M., Streeter, M., Mydlowec, W., Yu, J., and Lanza, G., *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*, Kluwer Academic, Dordrecht, Netherlands, 2003.

[19]Chryssolouris, G. and Subramaniam, V., “Dynamic scheduling of manufacturing job shops using genetic algorithms,” *Journal of Intelligent Manufacturing*, Vol. 12, No. 3, 2001.

[20]Mahfoud, S. and Mani, G., “Financial forecasting using genetic algorithms,” *Applied Artificial Intelligence*, Vol. 10, No. 6, 1996.

[21]Andreou, A., Georgopoulos, E., and Likothanassis, S., “Exchange-rates forecasting: A hybrid algorithm based on genetically optimized adaptive neural networks,” *Computational Economics*, Vol. 20, No. 3, 2002.

[22]Weismann, D., Hammel, U., and Back, T., “Robust design of multilayer optical coatings by means of evolutionary algorithms,” *IEEE Transactions on Evolutionary Computation*, Vol. 2, No. 4, 1998.

[23]Giro, R., Cyrillo, M., and Galvao, D., “Designing conducting polymers using genetic algorithms,” *Chemical Physics Letters*, Vol. 366, No. 1, 2002.

[24]Sambridge, M. and Gallagher, K., “Earthquake hypocenter location using genetic algorithms,” *Bulletin of the Seismological Society of America*, Vol. 83, No. 5, 1993.

[25]Benini, E. and Toffolo, A., “Optimal design of horizontal-axis wind turbines using blade-element theory and evolutionary computation,” *Journal of Solar Energy Engineering*, Vol. 124, No. 4, 2002.

[26]Lipson, H. and Pollack, J. B., “Automatic design and Manufacture of Robotic Lifeforms,” *Nature*, Vol. 406, No. 1, 2000.

[27]Nolfi, S. and Floreano, D., *Evolutionary Robotics*, MIT Press, Cambridge, Massachusetts, 2000.

[28]Wu, A. S. and Garibay, I. I., "Genetic algorithm optimization of life support system control," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 34, No. 3, 2004.

[29]Kortenkamp, D. and Bell, S., "Simulating Advanced Life Support Systems for Integrated Controls Research," *Proceedings International Conference on Environmental Systems*, 2003.

[30]NASA JSC, "JSC Advanced Life Support Requirements Document," Tech. Rep. JSC-38571C/CTSD-ADV-245C, NASA JSC (available at: http://advlifesupport.jsc.nasa.gov), 2003.

[31]Goudarzi, S. and Ting, K., "Top-Level Modeling of Crew Component of ALSS," *Proceedings International Conference on Environmental Systems*, 1999.

[32]Barta, D. J., Castillo, J. M., and Fortson, R. E., "The Biomass Production System for Bioregenerative Planetary Life Support Systems Test Complex: Preliminary Designs and Considerations," *Proceedings International Conference on Environmental Systems, SAE paper 1999-01-2188*, 1999.

[33]Jones, H. and Cavazzoni, J., "Top-Level Crop Models for Advanced Life Support Analysis," *Proceedings International Conference on Environmental Systems, SAE paper 2000-01-2261*, 2000.

[34]NASA JSC, "Advanced Life Support Baseline Values and Assumptions Document," Tech. Rep. JSC-47804/CTSD-ADV-484A, NASA JSC (available at: http://advlifesupport.jsc.nasa.gov), 2004.

[35]M. Edeen and D. Barta, "Early Human Testing Initiative Phase I," Tech. Rep. JSC-33636, NASA JSC, 1996.

[36]S. Brasseaux and J. Cornwell and L. Dall-Bauman and D. Henninger and B. Laws and D. Ming and C. Verostko and C. Bourland and P. O'Rear and K. Hurlbert, "Adanced Life Support Program Lunar Life Support Test Project Phase II Final Report," Tech. Rep. JSC-38800, NASA JSC, 1997.

[37]M. Edeen, "Lunar-Mars Life Support Test Project (LMLSTP) Phase III Final Report," Tech. Rep. JSC-29144, NASA JSC, 2000.

[38]Hanford, A., "Transient Modeling Challenge: A Lunar Reference Mission for a 90-Day Habitat," 2004.

[39]Object Managment Group, I., "Object Management Group," .

[40]Hanford, A. J., "Advanced Life Support Research and Technology Development Metric - Fiscal Year 2004," Tech. Rep. NASA/CR-2004-208944, NASA JSC (available at: http://advlifesupport.jsc.nasa.gov), 2004.

[41]Hanford, A. J., “Advanced Life Support Research and Technology Development Metric - Fiscal Year 2003,” Tech. Rep. NASA/CR-2004-208939, NASA JSC (available at: http://advlifesupport.jsc.nasa.gov), 2004.

[42]Hanford, A. J., “Advanced Life Support Research and Technology Development Metric - Fiscal Year 2002,” Tech. Rep. JSC-60313/CTSD-ADV-510A, NASA JSC (available at: http://advlifesupport.jsc.nasa.gov), 2003.

List of Table Captions

Attribute	Range	Scenarios 1	2	3	4	5
O_2 Injector Flow Rate	0–30 $\frac{mol}{hr}$	✓	✓	✓	✓	✓
OGS Power Consumption	0–2,500 W	✓	✓	✓	✓	✓
VCCR Power Consumption	0–100,000 W	✓	✓	✓	✓	✓
Power Production	0–700,000 W	✓	✓	✓	✓	✓
O_2 Store Capacity/Initial Level	0–2,000 mol	✓	✓	✓	✓	✓
Power Storage Capacity/Initial Level	0–500,000 W	✓	✓	✓	✓	✓
Food Storage Capacity/Initial Level	0–2,000 kg	✓	✓	✓	✓	✓
Crew Environment Volume	0–2,000,000 L	✓	✓	✓	✓	✓
Crop Environment Volume	0–2,000,000 L				✓	✓
Tomato Shelf Size	0–20 m^2		✓	✓	✓	✓
Lettuce Shelf Size	0–20 m^2		✓	✓	✓	✓
Integrated Water RS Power Consumption	0–3,000 W	✓	✓		✓	
Crew Water RS Power Consumption	0–3,000 W			✓		✓
Crop Water RS Power Consumption	0–3,000 W			✓		✓
Integrated Potable Water Storage Capacity/Initial Level	0–8,000 L	✓	✓		✓	
Crew Potable Water Storage Capacity/Initial Level	0–8,000 L			✓		✓
Crop Potable Water Storage Capacity/Initial Level	0–8,000 L			✓		✓
Integrated Grey Water Storage Capacity/Initial Level	0–4,000 L	✓	✓		✓	
Crew Grey Water Storage Capacity/Initial Level	0–4,000 L			✓		✓
Crop Grey Water Storage Capacity/Initial Level	0–4,000 L			✓		✓
Integrated Dirty Water Storage Capacity/Initial Level	0–5,000 L	✓	✓		✓	
Crew Dirty Water Storage Capacity/Initial Level	0–5,000 L			✓		✓
Crop Dirty Water Storage Capacity/Initial Level	0–5,000 L			✓		✓

Attribute	Attribute Unit	Weight	Weight Unit
O_2 Injector	mol/hr	2.1060×10^{-4}	$kg - hr/mol$
VCCR power req	W	0.0000E+00	kg/W
OGS power req	W	0.0000E+00	kg/W
Power production	W	6.2000E-02	kg/W
Crew/crop volume	L	1.3310E-01	kg/L
O_2 storage	mol	3.3088E-02	kg/mol
Power storage	W	6.8700E-01	kg/W
Food storage	kg	2.3600E+00	kg/kg
Tomato shelf size	m^2	3.7010E+01	kg/m^2
Lettuce shelf size	m^2	3.7010E+01	kg/m^2
WaterRS power req	W	0.0000E+00	kg/W
Potable water storage	L	1.0684E+00	kg/L
Grey water storage	L	1.0684E+00	kg/L
Dirty water storage	L	1.0684E+00	kg/L

List of Figure Captions

Figure 1: A comprehensive listing of the modules modeled by BioSim comprising an example life support system

Figure 2: System evolution is depicted as the genetic algorithm attempts more trials in configuration 1 run D.

Figure 3: The relationship between fitness and VCCR power consumption in configuration 1 run B. No concrete conclusion can be made regarding VCCR sizing from this output, as is the case with all replications of the GA with configuration 1.

Figure 4: Adequate food is critical for mission success as is shown here in configuration 2 run A, where a minimum food masses are related to mission length, and thus fitness.

Figure 5: Excessive power consumption is shown to be a limiting factor in enabling missions of extended duration. Diagram 'a' shows the relation of the crew water recovery system in configuration 3 run D. Diagram 'b' shows the same relation for the crop water recovery system in the same simulation.

Figure 6: Although power issues are critical, it is possible to identify minimum power expenditures to ensure a viable system in configuration 5 run D. However, a maximum threshold for power consumption also exists in this case.

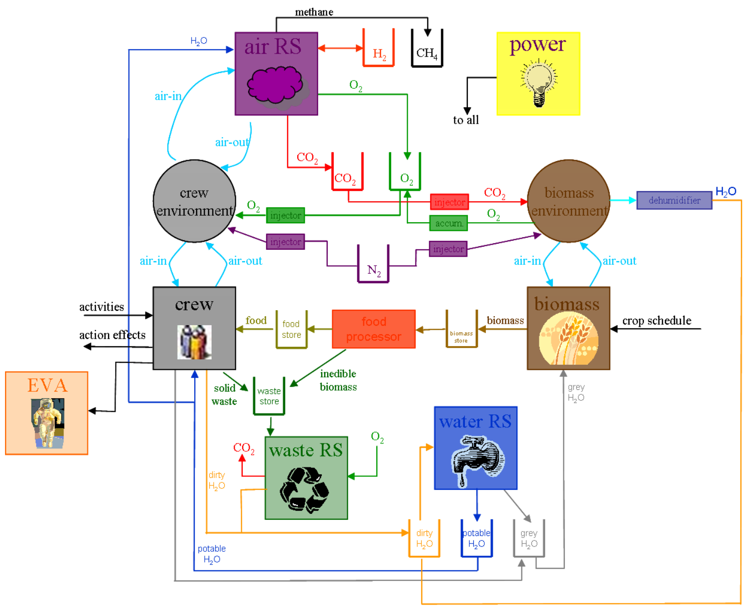
Figure 7: The fitness function is capable of identifying configurations which consume fewer resources. All configurations pictured complete the requisite 90-day mission. Evidently those systems with minimal power storage have higher fitness in configuration 2 run B.

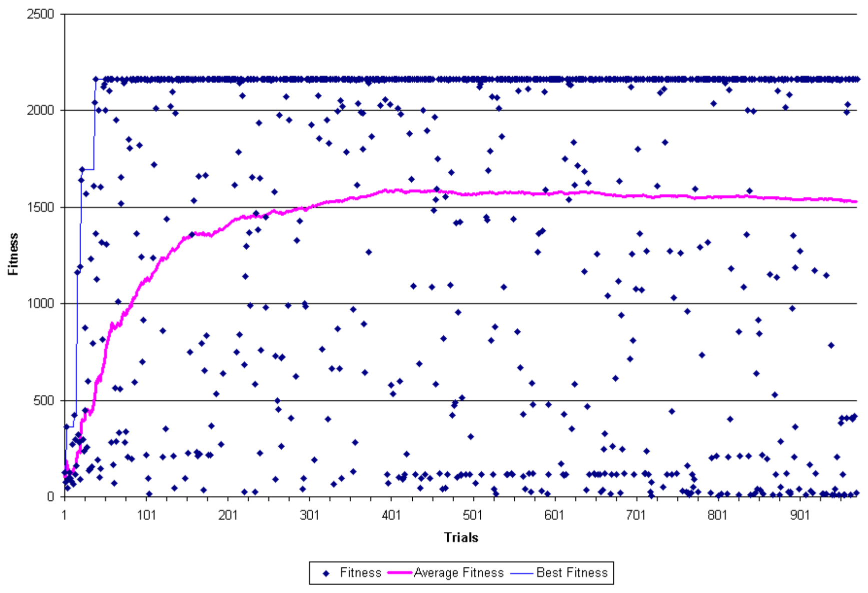
Figure 8: In configuration 4 run C, it can be seen that the VCCR can be sized very precisely. This is true for configurations 2-5.

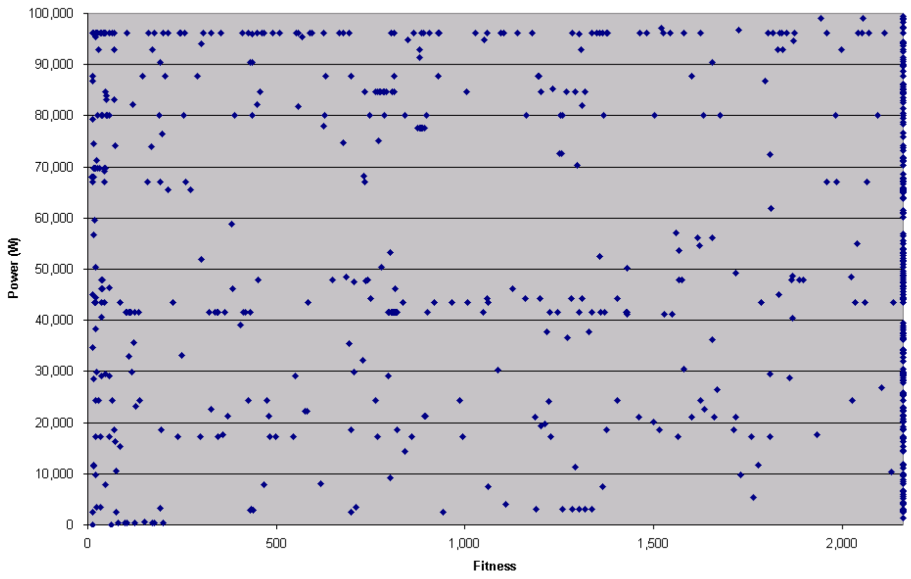
Figure 9: Sizing of various system components in each configuration is shown with 95% confidence intervals.

Figure 10: Average sizing and error bounds of the VCCR component in each configuration. The VCCR component was not precisely sized in configuration 1, but was very precisely sized in configurations 2–5.

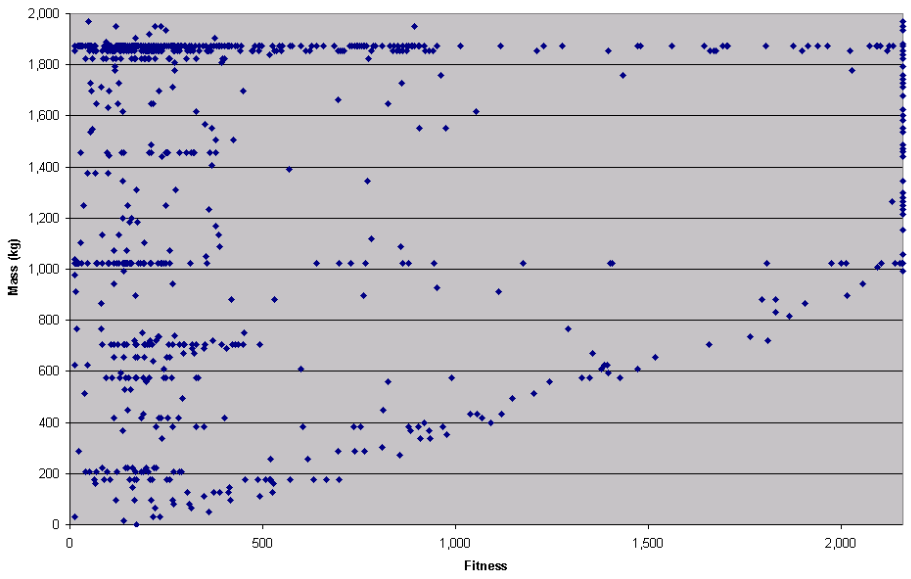
Figure 11: Fitness, being designed based upon ESM, can be utilized as a proxy for ESM. Each configuration is contrasted based upon this proxy.


methane
H_2O
air RS
H_2
CH_4
power
to all
air-in
O_2
air-out
CO_2
CO_2
O_2
injector
CO_2
biomass environment
dehumidifier
H_2O
crew environment
O_2
injector
accum.
O_2
injector
injector
N_2
air-in
air-out
air-in
air-out
crew
activities
action effects
food
food store
food processor
biomass store
biomass
biomass
crop schedule
EVA
solid waste
waste store
inedible biomass
grey H_2O
water RS
CO_2
O_2
waste RS
dirty H_2O
dirty H_2O
potable H_2O
grey H_2O
potable H_2O

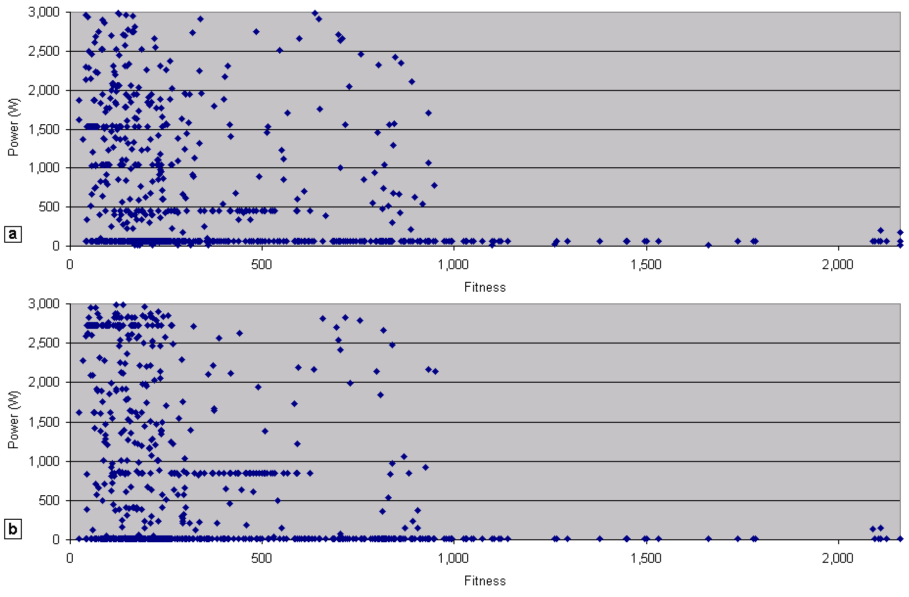

Fitness
Trials
2500
2000
1500
1000
500
0
1
101
201
301
401
501
601
701
801
901
Fitness
Average Fitness
Best Fitness



100,000
90,000
80,000
70,000
60,000
50,000
40,000
30,000
20,000
10,000
0
Power (W)
0
500
1,000
1,500
2,000
Fitness



2,000
1,800
1,600
1,400
1,200
1,000
800
600
400
200
0
Mass (kg)
0
500
1,000
1,500
2,000
Fitness


a
b
Power (W)
Fitness
0
500
1,000
1,500
2,000
2,500
3,000

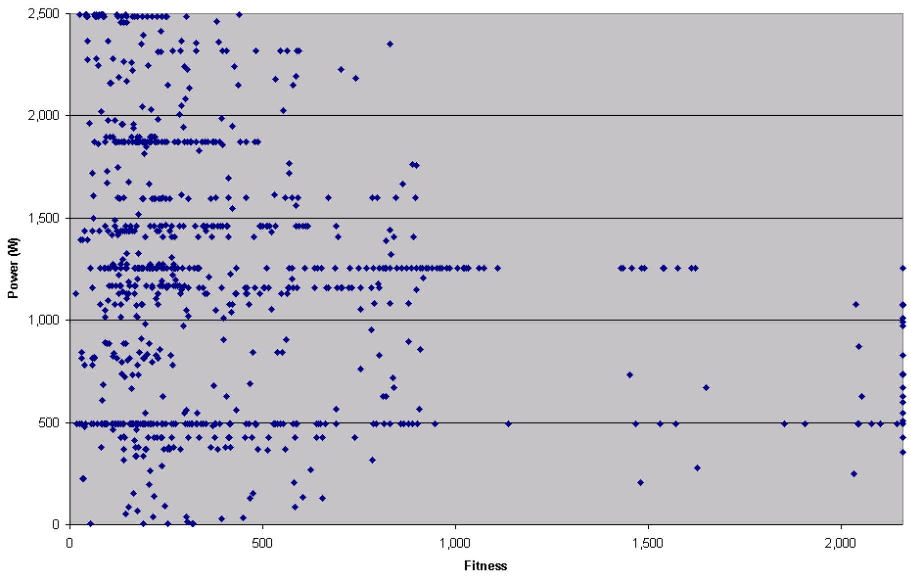
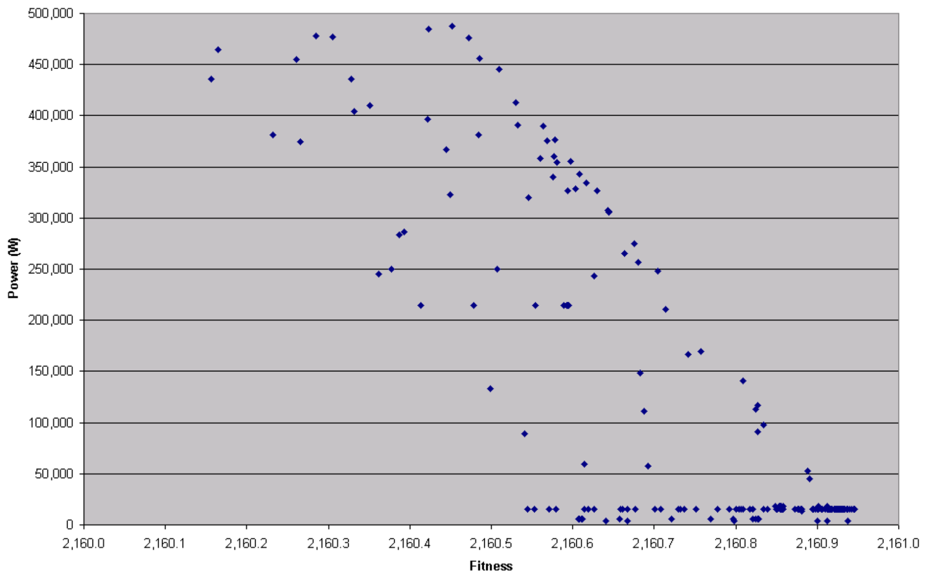
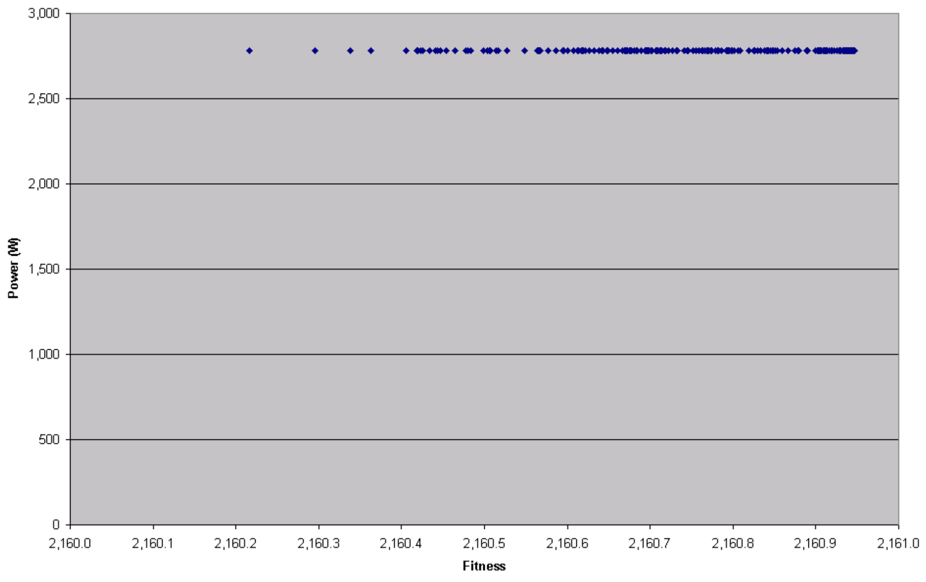


Figure 6:


Power (W)
500,000
450,000
400,000
350,000
300,000
250,000
200,000
150,000
100,000
50,000
0
2,160.0
2,160.1
2,160.2
2,160.3
2,160.4
2,160.5
2,160.6
2,160.7
2,160.8
2,160.9
2,161.0
Fitness


3,000
2,500
2,000
1,500
1,000
500
0
Power (W)
2,160.0
2,160.1
2,160.2
2,160.3
2,160.4
2,160.5
2,160.6
2,160.7
2,160.8
2,160.9
2,161.0
Fitness

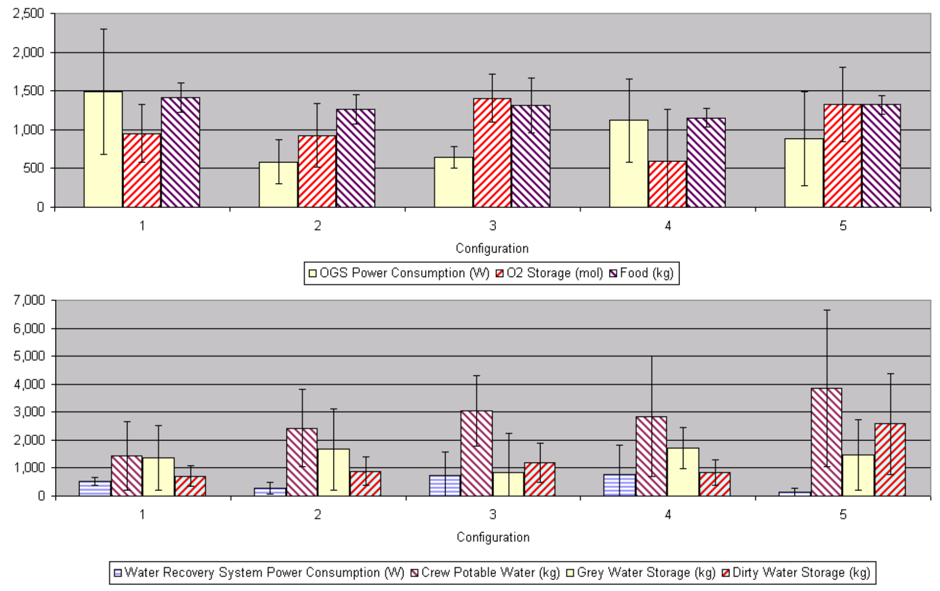
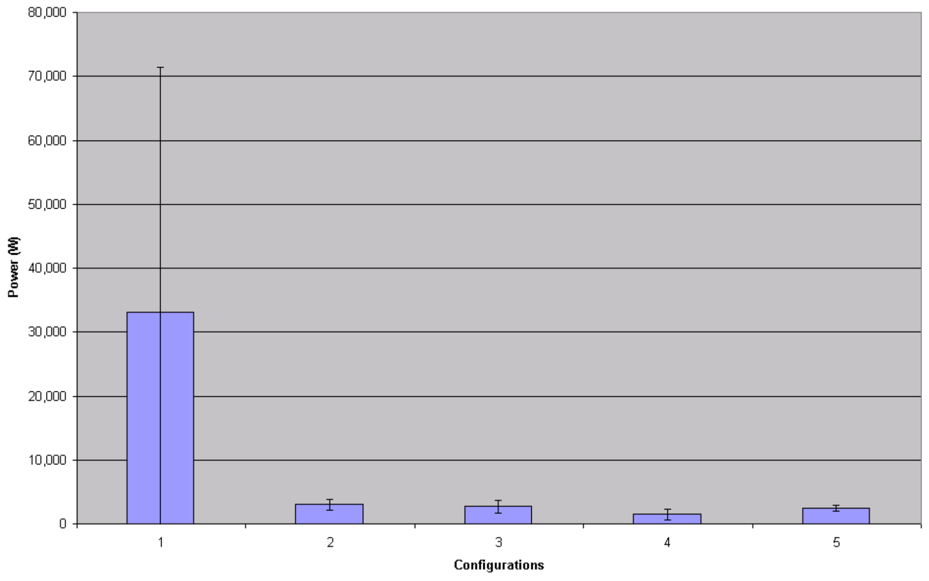
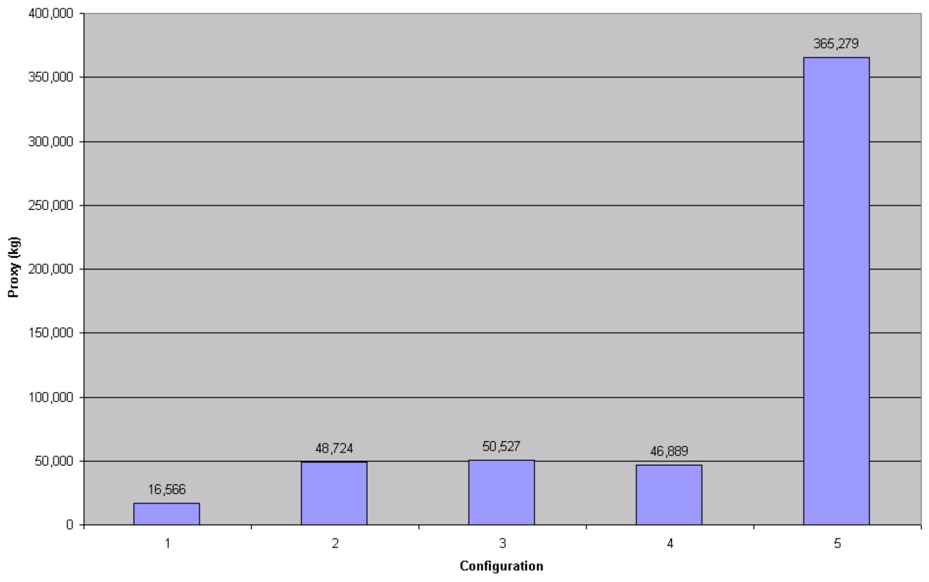

2,500
2,000
1,500
1,000
500
0
1
2
3
4
5
Configuration
OGS Power Consumption (W)
O2 Storage (mol)
Food (kg)
7,000
6,000
5,000
4,000
3,000
2,000
1,000
0
1
2
3
4
5
Configuration
Water Recovery System Power Consumption (W)
Crew Potable Water (kg)
Grey Water Storage (kg)
Dirty Water Storage (kg)

Figure 9:


80,000
70,000
60,000
50,000
40,000
30,000
20,000
10,000
0
Power (W)
1
2
3
4
5
Configurations



400,000
350,000
300,000
250,000
200,000
150,000
100,000
50,000
0
Proxy (kg)
16,566
48,724
50,527
46,889
365,279
1
2
3
4
5
Configuration