Edited "scrubbed" version

Automated Design of Spacecraft Systems Power Subsystems

Richard J. Terrile, Mark Kordon, Dan Mandutianu, Jose Salcedo, Eric Wood and Mona Hashemi
Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive
Pasadena, CA 91109
818-354-6158
rich.terrile@jpl.nasa.gov

Abstract—This paper[1,2] discusses the application of evolutionary computing to a dynamic space vehicle power subsystem resource and performance simulation in a parallel processing environment. Our objective is to demonstrate the feasibility, application and advantage of using evolutionary computation techniques for the early design search and optimization of space systems. With this approach, engineers specify several sets of conditional subsystem performance criteria to trade off subsystem goals of mass, cost, performance and risk. Once specified, the integrated evolutionary/simulation software will then automatically generate a design option for each criteria, selecting and sizing power elements based on the space system's anticipated performance in the simulated environment. Initial Activity plans from two actual JPL missions, Mars Exploration Rovers (MER) and Deep Impact (DI) are used to test the software. Our results have shown human-competitive advantages by generating credible design concepts much faster than humans are able to and without the need for expert initial designs.

TABLE OF CONTENTS

1. INTRODUCTION

This work describes the application of evolutionary computational techniques to the automatic optimization of spacecraft power sub-systems. Over the past three years the Evolutionary Computation Group at NASA's Jet Propulsion Laboratory (JPL) has had the objective of demonstrating the feasibility, application and advantage of using biologically inspired evolutionary computational techniques for the early design search and optimization of space systems. In general, we have demonstrated that the same computational tools used for computer aided design and for design evaluation can also be used for the automated optimization of designs [1]. These multi-parameter design simulators are run on cluster computers as a parallel population of designs with randomly varying input parameters and starting with a random initial designs. The results are competed and selected down to a smaller sub-set of parents that provide the basis (using genetic operators of mutation and gene cross-over) for the design parameters of the next generation. Given a large enough population, sufficient generations and the right conditions for evolution, we have demonstrated the feasibility of automatically optimizing simulated designs.

We applied these evolutionary techniques by incorporating the Multi-Mission Power Analysis Tool (MMPAT) into an evolutionary framework running in a parallel processing environment. This tool is a dynamic space vehicle power sub-system resource and performance simulation, and is one of several multi-mission design tools in use at JPL. These tools use the spacecraft activity plan to simulate uplinked commands over the mission duration. With this approach, engineers specify several sets of conditional sub-system performance criteria to trade off sub-system goals of mass, cost, performance and risk. Once specified, the integrated evolutionary/simulation software will then automatically generate a design option for each criteria, selecting and sizing power elements based on the space system's anticipated performance in the simulated environment.

In order to quantify the advantage of these techniques we compared our automatically generated power sub-system designs to two actual JPL spacecraft designs. Theses tests were run using activity plans from the Mars Exploration Rovers (MER) and Deep Impact (DI) missions. The MER activity plan is for a landed mission spending 90 sols (Martian days) on the surface. Deep Impact is a comet flyby spacecraft with an 8.3 month-long activity plan that includes cruise from 1.0 to 1.5 AU from the sun. Initial populations were created from randomly selected parameters and design requirements identical to MER and DI were specified.

2. DESIGN LIFE CYCLE

At the Jet Propulsion Laboratory (JPL) the life cycle of a deep space mission normally goes through six phases, each culminating with a review by project management and its funding agencies [2]:

1
[1] 0-7803-9546-8/06/$20.00© 2006 IEEE
[2] IEEEAC paper #528, Version 1, Updated Oct, 14 2005

- **Pre-Phase A:** Advanced Studies
- **Phase A:** Mission & System Definition
- **Phase B:** Preliminary Design
- **Phase C:** Design & Build
- **Phase D:** Assembly Test & Launch Ops
- **Phase E:** Operations

The process starts with Pre-Phase A where the goals and objectives of the mission are defined and several plausible mission concepts are created. These early mission concepts will trade off various elements in the design so that project managers can choose between different alternatives for mass, cost, performance and risk. Here a trade study is a process for seeking one or more optimal solutions when there are multiple, often conflicting, objectives. An optimal solution in this case means that if one objective improves, other objectives are compromised or traded off. The classic example of this is in car buying. Buyers must make a decision between cost and comfort since the less expensive cars are inevitably less spacious. This hypothetical trade-off is shown in Figure 1. To make select the design that best satisfies their requirements, the buyer would want to consider solutions that are evenly distributed along the Pareto-optimal front.

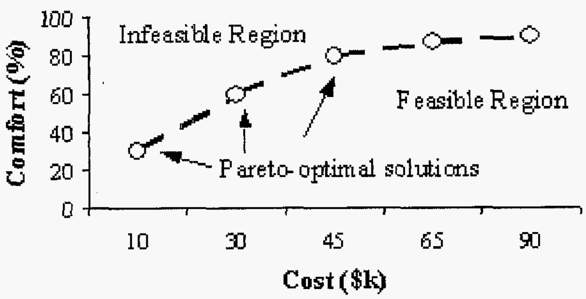


Figure 1 –Hypothetical Car Buying Trade-Off [3]

The cycle of goal definition, mission concept creation and design trade study is repeated many times in the early formulation phases. Each pass refines and improves the resolution of the design and removes design options from consideration. The product of this process is a single mission architecture characterized such that its effectiveness in achieving mission objectives can be properly evaluated. The mission architecture typically defines [4]:

- The Subject
- Orbit and Constellation
- Payload
- Flight System
- Launch Element
- Ground Element
- Mission Operations
- Command, Control and Communications Architecture

One important aspect of the mission architecture is the flight system. The purpose of the space vehicle flight system is to transport the payload safely to its destination and enable the return of science data to Earth. Typically the flight system is composed of several subsystems [2]:

- Power Subsystem
- Command & Data Handling Subsystem
- Telecommunications Subsystem
- Propulsion Subsystem
- Mechanical Subsystem
- Thermal Subsystem
- Guidance Navigation and Control Subsystem
- Spacecraft Flight Software

Each subsystem is responsible for a particular function, such as electrical power generation and distribution, and has design characteristics like solar array size, solar cell technology, secondary battery size and battery cell technology. Designing these subsystems to meet payload, trajectory, communication and activity requirements within the mass, cost and performance constraints of the project is vital for mission success. Automating this process would ensure consistent design quality while at the same time allowing experts to spend less time on routine tasks and more time evaluating various design options. This paper discusses an evolutionary computing approach for achieving the automated design of spacecraft systems.

With this methodology, evolutionary computing strategies are used with a dynamic, operations-validated power subsystem simulation to automate the design search and optimize space vehicle subsystem elements for a given set of project requirements and constraints. As we will demonstrate, this technique has several advantages over current approaches that rely on a small number of expert opinions employing worst-case estimates by generating credible power subsystem design concepts faster and for lower cost than humans are able to.

The paper begins with a brief overview of power subsystem design principles. It goes on to discuss the simulation used on this effort, the Multi-Mission Power Analysis Tool (MMPAT), and how it was integrated with an evolutionary computing framework. The paper continues by describing the fitness functions used on the effort and concludes by comparing the results generated to actual mission designs.

3. Power Subsystem Design

To properly develop a power subsystem fitness function it is important to understand the basic issues in spacecraft power subsystem design. Rather than consider all possible power sources it focuses on subsystems using photovoltaic solar power. Also, no consideration is given to cases where primary (non rechargeable) batteries might be used. After a brief discussion of some basic power subsystem design concepts, the various components are introduced in a logical order where later choices hinge upon the earlier selections.

Power Design Concepts

The main function of the electrical power subsystem is to generate and deliver electricity to all points of utilization on the spacecraft. The generated electricity must satisfy the power and energy requirements of the subsystems, be within the component ratings and voltage limits, as well as taking into account planned and unplanned usages of the spacecraft devices.

To handle interruptions in power generation, spacecraft power subsystems always provide an energy storage mechanism, such as a rechargeable secondary battery, for backup. To account for other unexpected problems, mission engineers establish flight rules to define the standard operating procedure for the spacecraft. Some of these rules add reserves such as power margin and energy margin to ensure reliable spacecraft operations. Others set critical thresholds such as minimum usable battery state of charge (SOC).

Power Margin

Power margin is the difference between the power generated and power consumed. When used as a flight rule, power margin establishes the minimum allowable power surplus at any point in time. This consideration is particularly important in cruise mode where it is necessary to keep the secondary battery fully charged in preparation for planned activities, solar obscurations or an unexpected contingency.

Energy Margin

Energy margin is the difference between the energy generated and the energy used during a period of time. When used as a flight rule, energy margin establishes the minimum surplus of energy during a 24 hour period. On landed missions there are often periods where no power is generated, so stored energy from a secondary battery must be used to keep critical systems operational. In this case, the power subsystem design needs to ensure that a positive energy balance can be maintained while performing typical daily activities so that a low power condition does not arise that would interfere with activities or worse, put the mission at risk during the night.

Usable State of Charge

A battery's available energy capacity, also known as state of charge (SOC), is measured as the amp-hours that the battery can deliver at the present discharge current before it reaches 100% depth of discharge. A battery is considered at a 100% depth of discharge when its cell voltage drops to a particular level. In spacecraft operations, the notion of SOC is modified to usable SOC so that engineers can monitor the energy available before the spacecraft begins to fail. Usable SOC measures the amp-hours that can be extracted from the battery before the bus voltage becomes less than the predefined minimum which would presumably cause the spacecraft to enter a low power state.

Usable SOC is computed by finding the "limiting SOC" that would necessarily exist at the present discharge current and the specified low bus voltage. The limiting SOC is then subtracted from the present SOC, giving the usable SOC.

Energy Storage – Secondary Battery

On a spacecraft, energy is typically stored in the secondary battery. There are three battery design considerations: the battery chemistry, bus voltage range and battery size. Selection of the battery chemistry is based on several factors: mass/volume, charge efficiency, thermal characteristics, need for battery charger and lifetime. Presently there are two battery chemistries commonly in use: Lithium-Ion (Li-Ion) and Nickel-Hydrogen (NiH2). Table 1 compares the characteristics of these two battery chemistries.

In practice, the charge efficiency, thermal characteristics and small mass/volume favor Li-Ion for landed missions while the longer lifetime and the absence of a charger/cell balancer circuit favor using NiH2 for space missions. For example, MER used the Li-Ion battery chiefly due to its low mass and compact size. DI used the NiH2 battery since volume constraints were not an issue and at the time that the DI design was done there was no flight experience with Li-Ion batteries.

The bus voltage is selected to work well with the devices powered by the bus as well as being compatible with the battery chemistry selected. Typical bus voltage ranges are from 24 to 36 volts to conform to the NASA 28 Volt Bus design standard. Conformance to the standard provides a host of off-the-shelf devices and designs.

Table 1. Battery Chemistry Comparison Chart

Battery Chemistry	Lithium-Ion	Nickel-Hydrogen
Mass efficiency	100 Watt-Hours/kg	30 – 60 Watt-Hours/kg
Volume efficiency	Relatively high	Relatively low
Charge efficiency	100%	Descends from 100% to 0% as SOC increases from ~70% to 100%
Thermal Characteristics	Moderate Thermal Load	Charge inefficiency (see above) becomes thermal load
Battery Charger/Cell Balancing Needed	Yes	No
Lifetime	2 - ?	Tens of years
Flight Experience	< 5	> 100

The bus voltage range and battery voltage are closely tied. Battery voltage is the product of cell voltage and the number of cells connected in series. Battery storage capacity is the product of cell capacity and the number of batteries connected in parallel. The bus voltage range and the number of battery cells in series should be selected so that at the maximum bus voltage the battery will attain a full or nearly full state of charge.

The battery size is calculated by evaluating periods during which loads exceed power generated (negative power balance). These periods can include night times for landed missions, solar obscurations for space missions as well as periods during which peak loads occur. The battery must be able to supply sufficient power during these periods to

Although batteries are custom made for each mission, in

practice Li-Ion batteries are commonly configured with 8 cells per battery and NiH2 are commonly configured with 22 cells per battery so that they operate well within the voltages dictated by the NASA 28 volt bus design.

Bus Voltage Control

As mentioned above, the bus voltage is selected to work well with the devices powered by the bus as well as being compatible with the battery chemistry selected. Typically, the bus voltage ranges are from 24 to 36 volts to conform to the NASA 28 Volt Bus design standard. The Bus Voltage Control (BVC) circuitry maintains the power bus within the

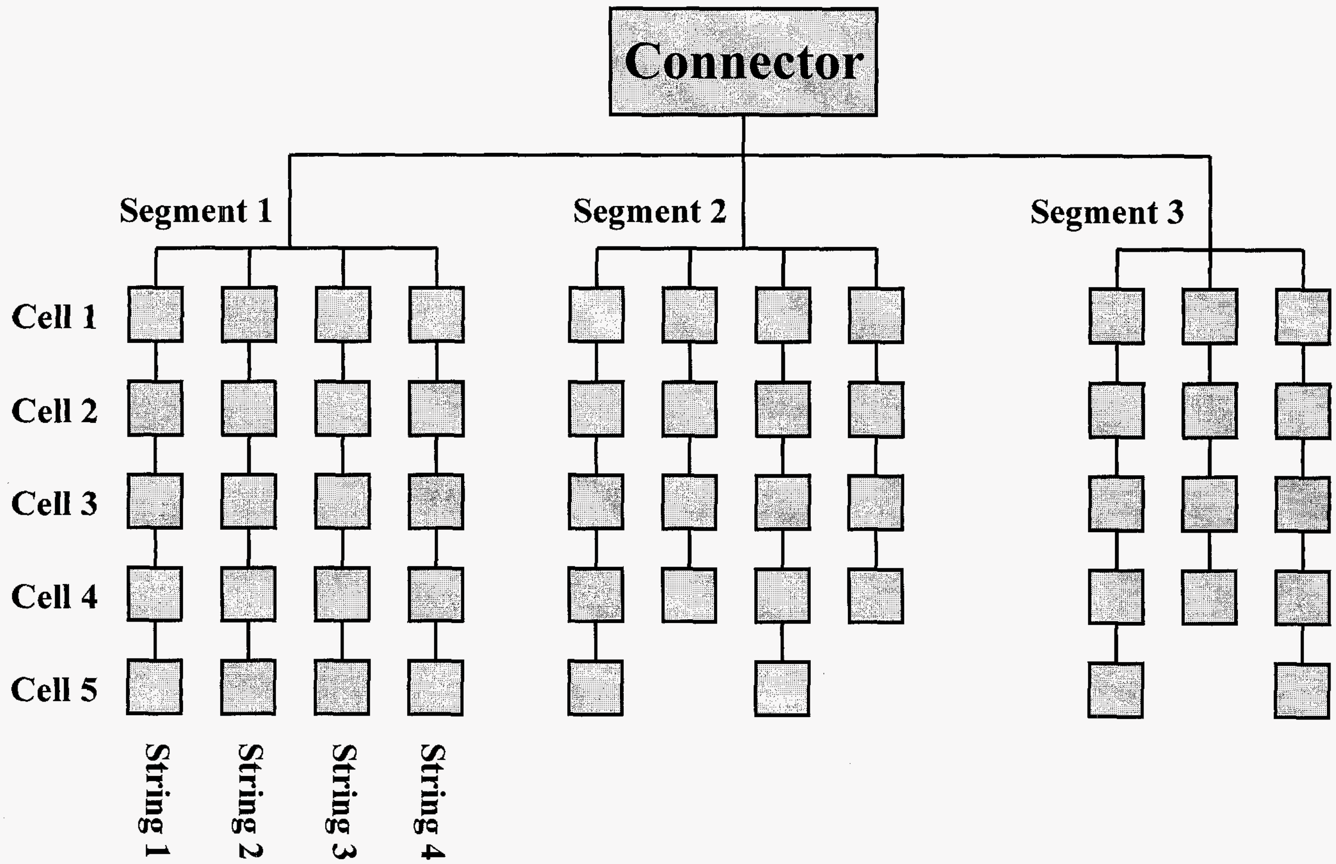


Figure 2. Example Solar Array Configuration

specified bus voltage range. On spacecraft the bus voltage prevent the bus voltage from dropping below the minimum.control varies greatly but one of three approaches is often used: shunt limiter, string switching and BVC circuitry.

The shunt limiter approach restricts the power bus from exceeding the maximum allowed bus voltage by directing sufficient current to shunt resistors when the maximum bus voltage is reached. In this case all solar array current is accepted from the array with the excess being shunted.

String switching on the other hand, makes use of a string selection matrix that allows the BVC circuitry to effectively switch in and out individual strings of solar cells, thus limiting the amount of current received from the solar array. The BVC circuitry can allow for reducing the battery charge rate set point or bus voltage set point as the battery voltage and temperature increase. This is commonly associated with the NiH2 battery due to its tendency to heat up as its SOC nears full charge.

The Deep Impact mission used the string switching method due to its low mass made possible by the lack of shunt radiators. MER used the shunt limiter method because of the simplicity of the circuitry required and the small shunt radiators needed since it was operating in a cold gas environment.

Power Generation – Solar Array

Solar arrays are divided into segments. Each segment consists of a set of parallel strings of solar cells. Within a string, the solar cells are connected in series. The segments are in turn joined in parallel at the connector to the power bus. Figure 2 illustrates an example solar array configuration.

The voltage produced by a single string is the sum of the voltages produced by the cells in that string. The current produced by a string is equal to the current produced by the weakest cell. The current produced by a segment is the sum of the currents produced by each string in the segment. The voltage produced by a segment is equal to the voltage produced by the strongest string. The current produced by the array is the sum of the currents produced by each segment. The voltage produced by the array is the voltage of the strongest segment.

The voltage produced by the solar array must be sufficient to fully charge the battery, that is, it must exceed the maximum bus voltage to a degree that will provide a sufficient charge current. The current produced by the array must be great enough to recharge the battery to prepare it for periods when loads exceed array power production. Solar cell current rises in direct proportion to the amount of solar insolation striking it. Solar cell voltage drops as cell temperature increases.

The implication of the above two effects are that the closer to the sun a spacecraft is, the fewer strings it needs to produce the necessary current. However, at the same time more cells per string are needed to produce the necessary voltage.

On a spacecraft whose mission requires significant changes in the distance from the sun, this can lead to designs with some segments having more cells than other segments. The segments with more cells produce enough current when close to the sun, but as the spacecraft becomes further from the sun, the shorter strings begin producing sufficient voltage due to the lowered temperature and thereby begin contributing current, just when the current from the long strings becomes insufficient due to the lowered insolation.

The Deep Impact mission is a good example of these concepts. DI used two string lengths: 44 strings with 22 cells per string and 112 strings with 16 cells per string. The configuration provided adequate current and voltage when near earth and also when at encounter (1.5 AU) when the cells were colder (more voltage) and receiving less insolation (less current).

Figure 3 shows an example solar array IV curve when near Earth at a sun distance of 1.0 AU. Figure 4 shows the IV curve when it encounters the comet Tempel 1 at 1.5 AU. The solar array voltage required the battery is 35 volts. The current required by DI to provide adequate power margin after launch and during early cruise was expected to be 16 amps. The current required near encounter was expected to be 21 amps, due to greater heating requirements and the needs of the various instruments.

One can see by examining the 35 volt points that the required currents are provided in both cases. The near Earth case depends only on the longer strings while the encounter case requires both the long and short strings.

The solar array configuration on the MER rovers uses a mix of string lengths: 15, 16 or 17 cells per string. This is due both to restrictions on the area available for solar cells and the existence of shadow-casting masts. A shadow on a cell effectively removes that cell's contribution to the voltage generated by the string. Without sufficient voltage, a string cannot generate current. The number of unshadowed cells needed in a string was 15. The extra cells in a string would allow it to contribute current even if one or two of its cells were in shadow.

4. Power Simulator

An important aspect of this approach is the use of a dynamic spacecraft power subsystem simulation that has been validated in actual mission operations. Using such a simulation ensures that our design search is based on the anticipated operation of the subsystem rather than human estimates. This section describes in detail the features we required, the simulator that was selected and how the simulator's results maps to mission flight rules. MER and DI design parameters are listed in Table 2

.

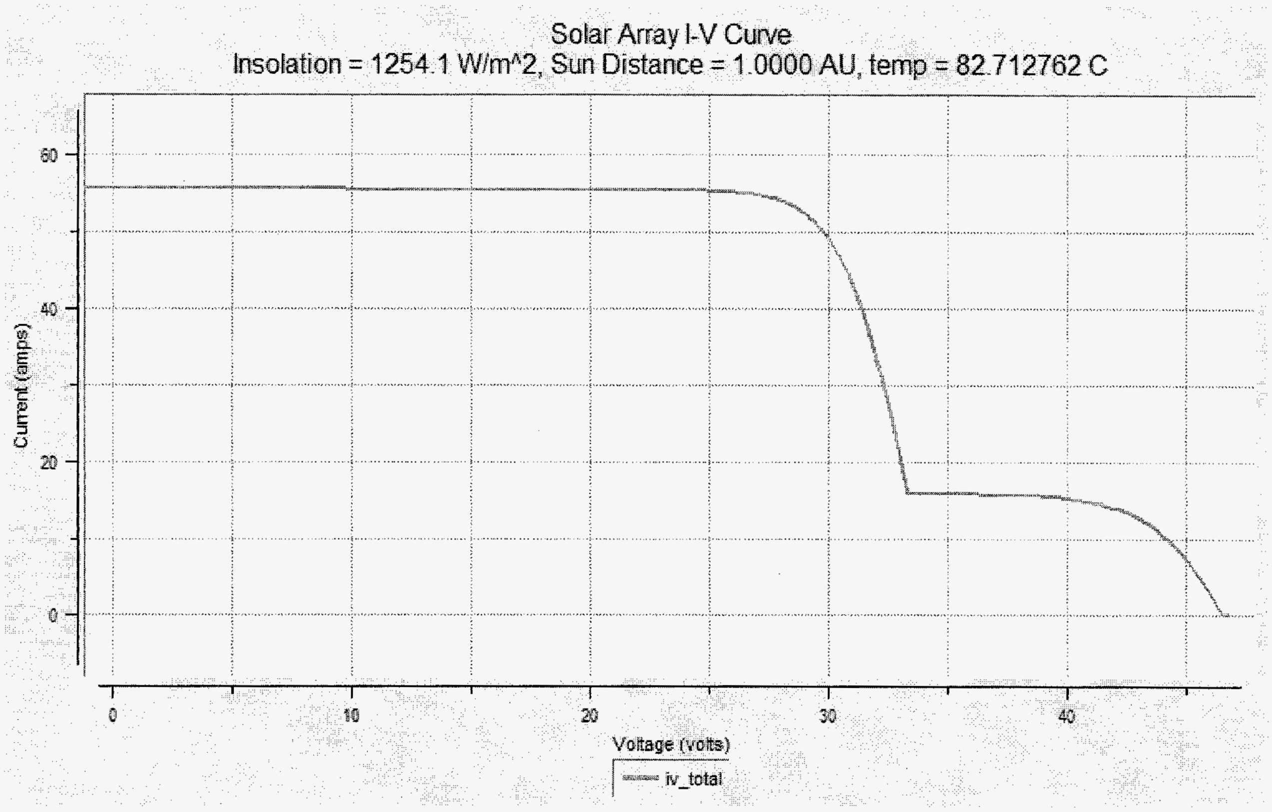


Figure 3. Example Near Earth Solar Array IV Curve

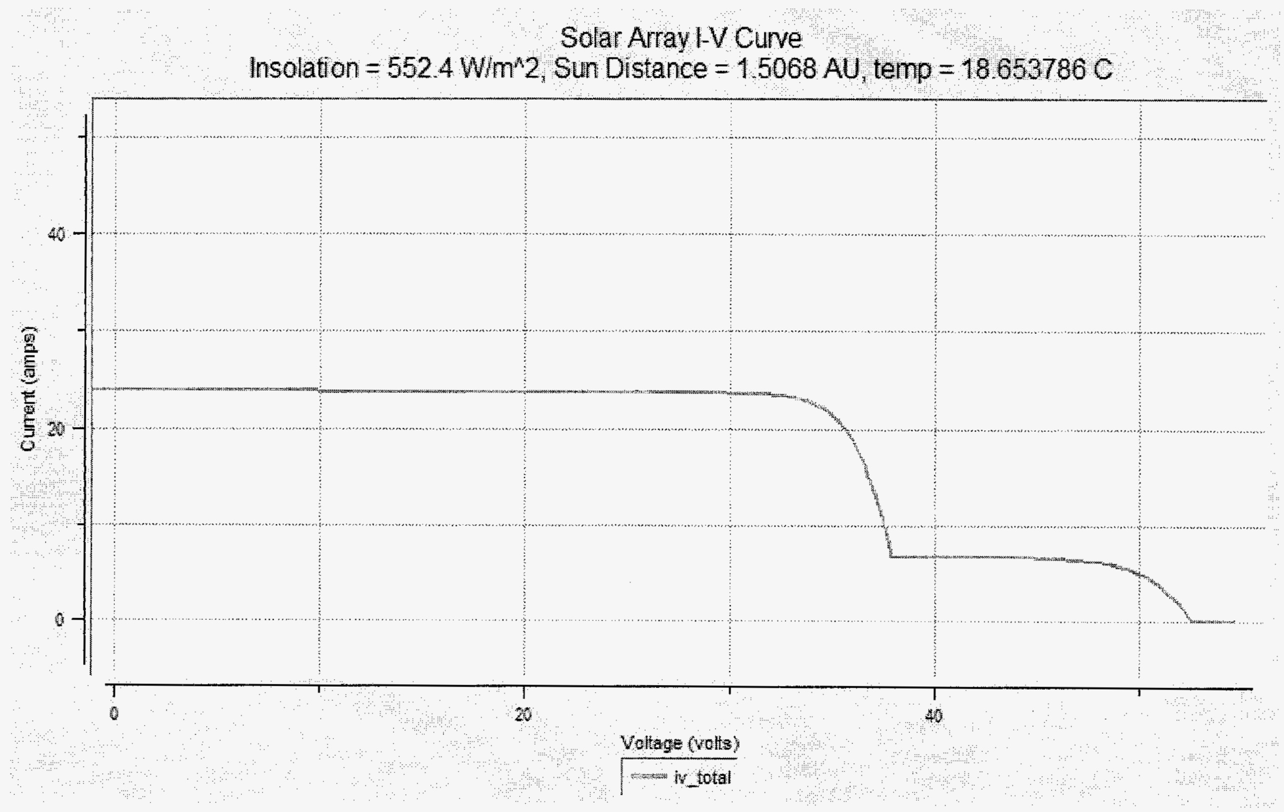


Figure 4. Example Encounter Solar Array IV Curve

Table 2 Summary of Deep Impact and MER Power Subsystem Flight Rules and Design Parameters

	MER Cruise	MER Surface	Deep Impact Cruise
Power Margin	Mc watts	n/a	D watts
Energy Margin	n/a	M watt-hrs	n/a
Usable SOC	Ma amp-hrs	Da amp-hrs	n/a
Total Number of Solar Array Segments	5	6	2
Total Number of Solar Array Strings	77	30	158
Total Number of Solar Cells	1310	480	2792
Battery Technology	Li-Ion	Li-Ion	NiH2
Battery Nameplate Capacity	20.0 amp-hr	20.0 amp-hr	16 amp-hr
Bus Control Method	Shunt Limiter	Shunt Limiter	String Switching
Bus Voltage	24-32.8	24-32.8	26.8-35
Mass of Solar Array Cell	0.01 kg	0.01 kg	0.01 kg
Cost of Solar Array Cell	$0.832k	$0.832k	$0.832k
Mass of Battery Cell	0.0167 kg/watt-hr	0.0167 kg/watt-hr	0.0167 kg/watt-hr
Cost of Battery Cell	$6.75k/amp-hr	$6.75k/amp-hr	$15.9k/amp-hr

Power Simulation Tool requirements

The task as defined required a simulation that could seamlessly handle multiple mission design alternatives and phases, and that could be integrated with an optimizer in a parallel processing environment. More specifically, the simulation needed to be a multiplatform library deployment with all of its design characteristics and state variables parameterized, and accessible through an Application Programming Interface (API). The API would also need to allow the user to enter an activity plan and trajectory.

Moreover, the simulation would need to use actual flight project data to quickly predict the resources and performance of the subsystem over the mission timeline, and would need to run in a closed loop manner with environment models that were, preferably, already integrated. Lastly, while not specifically required for this task, we wanted the simulation to be able to respond dynamically to inputs from other subsystems for compatibility with future research efforts.

Power Simulation in Operations

Ideally our simulator would be validated in operations. These operations tools have some unique input and output considerations in that each simulation must be able to input the design of their subsystem as well as a time-ordered sequence of events known as an activity plan. This plan is generated daily to determine what the spacecraft is intended to do. From this plan, a sequence is built and uploaded as instructions to the spacecraft.

To ensure that activity plans do not over commit resources and jeopardize the mission, the simulators must be able to predict what the resources will be after running the plan and verify that they are within the flight rule margins.

For example on MER, every sol the rover's battery state of charge and other key pieces of telemetry were supplied to a power simulator used by the mission planners to predict power generated, loads and the battery state of charge. Using this tool, they would then develop a sequence of rover commands that would include as much science and other useful activity as possible while still maintaining the required energy margin.

On Deep Impact, power analysts ran a power simulation that considered all expected loads plus all thermostatically-controlled heaters before each TCM. Power generation was simulated considering the sun distance and range of sun angles to be encountered during the maneuver. Results of the simulation were examined to be sure that the specified power margin would be maintained throughout the maneuver

Multi-Mission Power Analysis Tool

Given these requirements and the fact that we wanted a proven operations tool, we choose to use the Multi-Mission Power Analysis Tool (MMPAT) used on MER and Deep Impact. MMPAT is one tool in a suite of Multi-Mission Subsystem Analysis Tools at JPL [5]. It is a multiplatform software simulator currently used in Mars Exploration Rover (MER) operations to predict the performance and resources of space vehicle electrical power subsystems before a sequence of activities is uploaded.

The simulation can provide variable fidelity and produces dynamic time and sequence dependent results rather than static point solutions. As such, it models the behavior of power sources and energy storage devices as they interact with the spacecraft loads and the environment over a mission timeline at a level of detail appropriate to each stage of the project lifecycle, which in MER's case, is operations. The models in MMPAT include:

- Solar Array Model
- Solar Array Thermal Model
- Orbital Mechanics
- Astrodynamics Model
- Pointing Model
- Atmospheric Model
- Secondary Battery Model
- Secondary Battery/Thermostatically Controlled Heater Thermal Model
- Power Bus Model
- RTG Model
- Power Equipment List Model

All of the models were developed by power subsystem experts or adapted from validated heritage models. The tool itself comes with models for many of the most commonly used power sources, storage devices and power bus control methods used on space vehicles today. All of these models have been validated on previous or current missions, such as Pathfinder and MER, and give an accurate prediction of the system performance and resources.

The simulation is controlled by model parameters and was designed to be data-driven, modular and multiplatform. This means the models can be expanded to include additional hardware types. It also means that the application can be deployed stand-alone or as a library in another application, which in our case means integrated with an optimizer in a parallel processing environment. Moreover, the parameterized interface on MMPAT can also be used to change the mission type and analyze different mission phases since the tool supports the analysis of planetary landers, planetary orbiters, heliocentric orbiters and rovers as well as cruise, landed and orbiting phases and special events like flyby, TCM and EDL.

MMPAT Flight Rule Outputs

In some case MMPAT's outputs map directly to the flight rules, in other cases they do not. The usable SOC maps directly as the MMPAT variable: batt_usable_amphrs. However, while the MMPAT variable Emargin measures the energy margin of the spacecraft in watt-hours it accumulates the value over the lifetime of the mission. In ordered to compare this value to the daily energy margin flight rule, we needed to subtract the energy margin at the beginning of the day from the end-of-day value.

For power margin, MMPAT calculates the power margin in watts based on power actually taken off the solar array. Normally this would be fine except in the string switching algorithm, strings can be switched off. For a better comparison to the flight rule we want to use the power that could be generated from the solar array. The equation in this case was:

power margin = sa_Pavail - sa_Pactual + Pmargin

where:

sa_Pavail = Power available on array
sa_Pactual = Power being delivered by array
Pmargin = margin computed by MMPAT

5. SYSTEM ARCHITECTURE

With the flight rules defined and a power subsystem simulator selected, we now needed an automated method to search through the design space for different sizes and combinations of power equipment. We also need an architecture that allowed us to test the performance of each design using the simulation. The main architectural driver was the choice of optimization strategy. Since we wanted to find multiple optimal solutions in a single run, evolutionary computing was the natural choice. This section describes the evolutionary computing strategy and the resulting system architecture.

The Evolutionary Computing Strategy

Evolutionary computing seeks to generate optimal or near-optimal solutions for a given system by using a computer program to simulate the biological processes of natural selection [6,7]. This means that by using a process of random variation and selection through competition in an environment, the quality of solutions will iteratively improve. Simply put, the process involves generating a population of candidate solutions, evaluating how well they satisfy the requirements and constraints, and then randomly mating the solutions to create children for the next generation. The selection of mates is weighted toward the better solutions so that they will have a reproductive advantage. Implicit in this process is the notion of a particulate mechanism of inheritance.

In biology, organisms have a genetic coding known to as a genotype. Their morphology, physiology and behavior are referred to as the phenotype [8]. They are related to each other in that an organism's genotype describes influences and controls its phenotype. This means that changing an organism's genes will change its function, structure or behavior, and will oftentimes affect several characteristics at once since genes are typically pleiotropic. So in our application the design parameters are the genotype of the system, which succinctly describe and influence the structure and behavior of the subsystem or phenotype. Reproducing in this instance means contributing some design parameters from each parent to the child thus creating a combination of both of them that is hopefully better. This evolutionary process continues until some number of iterations has occurred or until the solution converges.

To support trade studies the system needed to be able to simultaneously generate multiple diverse solutions rather than a single point solution. This is achieved by allowing the user to create population slots that are used to bin segments of the population. Each slot is defined by a conditional that sets the membership criteria. The conditional is patterned after spacecraft flight rules for power margin, energy margin, usable SOC, and has the following form:

- Not less than x units

Where power margin units are in watts, energy margin units are in watt-hours and usable SOC is in amp-hours.

Typically cruise mission phases use power margin and landed missions use energy margin, but users can concatenate conditionals if desired. For example, a filter may be specified as 'not less than 20 watts and not less than 5.5 amp-hrs'.

The conditional may be modified by another conditional to bound the range of the slot. This optional conditional is in the form of:

- Not greater than x units

The conditional may also be modified by an exception that ignores the conditional for a certain period of time. The exception is in the form of:

- Except from t0 to t1

Exceptions may be concatenated. For example, to set a slot for a cruise mission with a trajectory correction maneuver that temporarily points its solar arrays away from the Sun the user could indicate 'not less than 20 watts except from day 10 to day 11 and not less than 5.5 amp-hrs'.

In this way the users may emphasize and any number of multiple solutions simultaneously.

PGASIM

The evolutionary mechanism described above is relatively straightforward to implement in software, but since there are numerous genetic algorithm frameworks on the market today we decided to use one that was already available. After a brief survey, we choose PGAPack developed by David Levine of the Mathematics and Computer Science Division at Argonne National Laboratory [9].

PGAPack is a general-purpose, data-structure-neutral, parallel genetic algorithm library. It is intended to provide most capabilities desired in a genetic algorithm library, in an integrated, seamless, and portable manner. The package consists of a set of library routines that supply the user multiple levels of control over the optimization process. The levels vary from default encodings, with simple initialization of parameters and single statement execution, to the ability to modify all relevant parameters in the optimization process at a low level. User written routines for evaluation or crossover and mutation can also be inserted if necessary.

Because the calculations of the fitness function involve computations that can be quite intensive, executing the evolutionary computing algorithm on massively parallel computers is essential for high-fidelity models. PGA Pack supports this by using the Message Passing Interface (MPI) for parallel execution on a number of processors. Thus the primary advantages that this package had over others is that it executed on cluster systems and is open source.

PGAPack did require some modifications for use in our application. Our changes to the package, which we are calling PGASIM to distinguish it from the original, were as follows:

1. Changed the code from C to C++.
2. The user provided functions are now methods in user defined classes.
3. A comprehensive configuration file is driving the algorithm instead of the user making explicit calls to set the parameters.
4. Increased flexibility by creating more place holders for custom code.
5. Enhance features like multi-criteria optimization and hybrid criteria.

Design Generation Process

At the beginning of an optimization run the head node reads the PGA and Optimization configuration files. This information is then used to instantiate the genetic algorithm data structures and create the initial population of power subsystem designs using randomly generated design parameters.

Once the initial population has been created, the compute nodes are configured by the head node to execute the simulation. Each compute node reads a reference MMPAT configuration and alters it according to the specific parameter values for that member. It also reads in the activity plan file, which is the same for all individuals, and executes the simulation.

After all of the members of a population have been simulated, the results are evaluated by the head node. First a check is made to determine if the design encountered any simulation errors that indicate if the design is infeasible, such as when the usable state of charge drops to or below zero. Then the output results from the simulation are examined and the member is put into a slot that satisfies the conditional. The ones that do not fit any slot are discarded. The members of each slot are then ranked from lowest to highest using the ranking criteria. Currently, there are two ranking criteria that can be specified by the user, either cost or mass.

The mass of the power subsystem is determined using a linear model. The mass of the power subsystem is obtained by determining the mass of the solar array and the mass of the battery. The solar array mass is determined by calculating the total number of solar cells and multiplying that value by mass-per-cell constant. Once this value has been calculated, the mass of the battery is determined by multiplying the battery's nameplate capacity by the mass-per-watt-hours constant. The sum of these two masses, determined the total mass of the power subsystem.

A similar model is used to determine the cost of the power subsystem. The cost of the solar array is calculated by multiplying the total number of cells with by a cost-per-cell constant; the cost of the battery, calculated by multiplying the battery's nameplate capacity by a cost-per-watt-hours constant.

When the entire population has been slotted and ranked, the head node collects the simulation results from the compute nodes and checks the stopping condition to decide whether the process should continue. If the stopping condition has been met, a report is generated containing the design parameters of the best population members. Otherwise the head node generates a new population.

The top members (lowest cost or mass) are selected to continue to the next generation so as not to lose the best candidates. The number of elites that are preserved for the next generation is calculated by subtracting the population replacement size from the population size. No duplicates are allowed in this elite survival list so if the same member exists at the top of another slot it is ignored and the next best candidate is selected. The rest of the replacement population is generated using a tournament operator to select parents and creating children by varying the parent's design parameters with crossover and mutation. After the new population has been generated, the cycle continues.

6. TEST CASES

Two different mission scenarios were selected to test this automated design system, the MER surface phase and Deep Impact cruise phase. The intent was to test the system using different phases on different missions and compare the generated solutions to the actual mission design. Normally we would use a more idealized formulation phase activity plan for the optimization, but for a better comparison to the design, we used an actual activity plan from mission operations for both test cases. While this provided a good basis for comparison it also caused some problems because missions will occasionally violate their own flight rules. Because of this, we needed to mask out times when this occurred.

The optimization was run on two separate clusters to test system performance. The first is a Dell distributed memory parallel processor system with 3.2 GHz Intel Pentium 4 Xeon processors. The second is a Beowulf cluster with 450 MHz Intel processors. This section describes the tests cases and the results.

MER Surface Mission Test Case

MER Spirit Rover was used for the surface mission test case. The rover was placed in its actual location of 14.95 degrees south latitude and given an actual 90 Sol activity plan from mission operations. This corresponds to the original planned length of surface operations of MER-A at the Gusev Crater landing site. The design parameters that we were interested in generating and their MER values included:

Total Number of Solar Array Strings: 30
Total Number of Solar Cells: 501
Battery Technology: Li-Ion
Battery Nameplate Capacity: 20.0 amp-hrs
Bus Control Method: Shunt Limiter

For this optimization the following intervals were used for each of the variable design parameters:

Number of Strings : 1 to 40
Number of Cells per String : 1 to 40
Nameplate Capacity in amp: 1 to 30

The following cost and mass values were used and remained fixed throughout the analysis:

Solar cell cost: $0.832k
Solar cell mass: 0.01 kg per cell
Battery cost: $6.75k/amp-hr
Battery mass: 0.01667 kg/watt-hr

Using our system linear models to compute the total mass and cost of the Spirit configuration, we derived the following values:

Mass of the SA & Battery: 5.29 kg
Cost of the SA & Battery: $551.83k

Since this was a surface mission only the energy margin (EM) and usable SOC (USOC) flight rules were used. The following slots were defined as:

Slot 1: EM > M watt-hrs AND USOC > Ma amp-hrs
Slot 2: EM > 0 watt-hrs AND USOC > Ma amp-hrs
Slot 3: EM > 2*M watt-hrs AND USOC > Ma amp-hrs
Slot 4: EM > M watt-hrs AND USOC > 2*Ma amp-hrs
Slot 5: EM > M watt-hrs AND USOC > 0.2*Ma amp-hrs
Slot 6: EM > 2*M watt-hrs AND USOC > 2*Ma amp-hrs
Slot 6: EM > 0 watt-hrs AND USOC > 0.2*Ma amp-hrs

Time masks, in decimal hours, were defined to account for start up and late mission where flight rules were violated to add more science collection.

Surface Mission Results

Table 3 and 4 shows the designs generated on the Dell cluster after 200 and 400 generations, respectively, The first column shows the MER design parameters we were concerned with, while the remaining columns show the generated design parameters with slot 1 duplicating the MER power subsystem flight rules.

The first issue that needed to be addressed was the values of evolutionary parameters. Based on our experience we chose appropriate mutation and crossover probabilities on a population size of 128. Convergence of the design parameters using these evolutionary values occurred somewhere between 200 and 400 generations.

Table 3 MER Surface Mission Power Subsystem Designs after 200 Generations

	MER	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7
Energy Margin (watt-hrs)	> M	> M	> 0	> 2M	> M	> M	> 2M	> 0
Usable SOC (amp-hrs)	> Ma	> Ma	> Ma	> Ma	> 2Ma	> 0.2Ma	> 2Ma	> 0.2Ma
Total Number of Solar Array Strings	30	32	30	42	32	32	42	30
Total Number of Solar Cells	501	458	432	614	458	458	614	432
Battery Technology	Li_Ion	Li_Ion	Li_Ion	Li_Ion	Li_Ion	Li_Ion	Li_Ion	Li_Ion
Battery Nameplate Capacity (amp-hrs)	20.0	26.62	26.62	18.82	26.62	26.62	18.82	26.62
Mass of SA & Battery (kg)	5.28	5.02	4.76	6.45	5.02	5.02	6.45	4.76
Cost of SA & Battery ($k)	551.83	560.71	539.08	637.92	560.71	560.71	637.92	539.08

Table 4 MER Surface Mission Power Subsystem Designs after 400 Generations

	MER	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7
Energy Margin (watt-hrs)	> M	> M	> 0	> 2M	> M	> M	> 2M	> 0
Usable SOC (amp-hrs)	> Ma	> Ma	> Ma	> Ma	> 2Ma	> 0.2Ma	> 2Ma	> 0.2Ma
Total Number of Solar Array Strings	30	34	29	38	34	34	38	29
Total Number of Solar Cells	500	477	417	533	477	477	533	417
Battery Technology	Li_Ion	Li_Ion	Li_Ion	Li_Ion	Li_Ion	Li_Ion	Li_Ion	Li_Ion
Battery Nameplate Capacity (amp-hrs)	20.0	22.42	26.62	22.21	22.42	22.42	22.21	26.62

Mass of SA & Battery (kg)	5.28	5.14	4.61	5.70	5.14	5.14	5.70	4.61
Cost of SA & Battery ($k)	551.83	548.22	526.60	593.38	548.22	548.22	593.39	526.60

As can be seen, diverse solutions can be generated by varying the values in the flight rules. The slot with the lowest margin, and highest risk, generated the least expensive solution and the slot with the highest margin was the most expensive. Slot 1 where the MER flight rules were used generated design parameters that were less expensive and within 10% of the MER design. The exception was the design parameters for the number of strings which was larger than the MER design. This is probably because we did not use a flight rule or requirement to address the distribution of cells on the strings. Introducing a requirement for the voltage and/or current at the solar array connector would likely fix this problem.

Using 128 nodes, 400 generations of this 90 day activity plan took 7 hours to complete on the small clusters and less than one hour on the large cluster using 1000 nodes.

Deep Impact Cruise Mission Test Case

The Deep Impact mission was used for the cruise mission test case. The spacecraft trajectory starts from Earth or 1.0 Astronomical Units (AU) and travels an ellipse to 1.5 AU where it will encounter Comet Tempel 1. It took approximately 8.3 months to traverse this distance.

The activity plan was from DI operations. During the mission there were five trajectory correction maneuvers where the solar array edge-on to the Sun. This had the effect of forcing the battery to be the sole source of power to the spacecraft during this time.

The design parameters that we were interested in generating and their DI values included:

Total Number of Solar Array Strings: 158
Total Number of Solar Cells: 2792
Battery Technology: NiH2
Battery Nameplate Capacity: 16.00 amp-hrs.

For this optimization the following intervals were used for each of the variable design parameters:

Number of Strings : 1 to 40
Number of Cells per String : 1 to 40
Nameplate Capacity in amp: 1 to 30

The following cost and mass values were used and remained fixed throughout the analysis:

Solar cell cost: $0.832k
Solar cell mass: 0.01 kg per cell
Battery cost: $19.3k/amp-hr
Battery mass: 0.01667 kg/watt-hr

Using our system linear models to compute the total mass and cost of the DI configuration, we derived the following values:

Mass of the SA & Battery: 28.19 kg
Cost of the SA & Battery: $2361.82k

Table 5. Deep Impact Cruise Mission Power Subsystem Designs after 239 Generations

	DI	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
Power Margin	D	> D	> 0	> 0.5*D	> 1.5*D	> 2*D	> 2.5*D
Total Number of Solar Array Strings	158	169	154	159	182	188	204
Total Number of Solar Cells	2792	2862	2676	2712	3218	3371	3830
Battery Technology	NiH2	NiH2	NiH2	NiH2	NiH2	NiH2	NiH2
Battery Nameplate Capacity (amp-hrs)	16.00	11.35	11.56	11.35	12.48	14.88	24.29
Mass of SA & Battery (kg)	28.19	28.81	26.95	27.31	32.39	33.96	38.71
Cost of SA & Battery ($k)	2577.94	2561.58	2410.23	2436.78	2875.87	3041.23	3572.84

Since this was a surface mission only the power margin flight rules was used. The following flight rules were used:

Power Margin: D watts

The following slots were defined:

Slot 1: Power Margin > D watts
Slot 2: Power Margin > 0 watts
Slot 3: Power Margin > 0.5*D watts
Slot 4: Power Margin > 1.5*D watts
Slot 5: Power Margin > 2*D watts
Slot 6: Power Margin > 2.5*D watts

Time masks, in decimal hours, were defined to account for trajectory correction maneuvers and the release of the impactor.

Cruise Mission Results

Table 5 shows the designs generated on the Dell cluster after 293 generations. The first column shows the DI design parameters we were concerned with, while the remaining columns show the generated design parameters. Slot 1 is identical to the DI power subsystem flight rules.

Using 51 nodes, 293 generations of this 8.3 month activity plan took 12 hours to complete on the small clusters and less than one hour on the large cluster using 1000 nodes..

7. SUMMARY OF RESULTS

Results indicate that credible design concepts for spacecraft power sub-systems can be generated in an hour after about 20,000 MMPAT evaluations. Evolved designs showed slightly better cost and mass performance and were automatically generated as a trade study in a less than one hour. Multiple diverse designs with different mass, cost, performance and risk postures were generated providing project management and its funding agencies several plausible design concepts to choose from. These results demonstrate human-competitive advantages by generating credible design concepts faster that humans are able and without the need for initial design requiring domain expertise.

8. CONCLUSIONS

It takes a team of experienced JPL domain experts at least two weeks to generate a credible pre-award mission concept [10]. Results obtained using two different mission scenarios in different mission phases indicate that credible design concepts for a spacecraft power subsystem can be generated in an hour using evolutionary computing and a dynamic space vehicle power subsystem resource and performance simulation in a parallel processing environment. Moreover, multiple diverse designs with different mass, cost, performance and risk postures can be generated providing project management and its funding agencies several plausible design concepts to choose from. The results also seem to indicate that the speed of automated design generation appears to be driven primarily by the length of the mission to be evaluated and speed of the processing hardware, rather than the number of design parameters to be optimized.

Of course, several components are required to achieve good results:

1. System flight rules and/or technical requirements
2. Time-ordered list of S/C trajectory and state changes
3. A validated system simulation capable of ingesting time-ordered events, design parameters and outputting relevant system state variables
4. An optimization strategy
5. A parallel processing environment
6. A parallel- processing integration environment

Nevertheless, as promising as the results are there is still much work to be done before this system can be used in the flight project design process. On the implementation side, this approach also needs to be applied to other flight system subsystems. Once completed all of the subsystems need to be integrated together to provide more comprehensive design solutions.

In addition, it is likely that more power subsystem design parameters such as battery and solar cell technology will need to be optimized. It is important that when such variables are added, the simulation model supports the tradeoffs to be made and that a flight rule and/ or technical requirement is available that can be checked.

On the theoretical side, it will be important test the system against other missions. Doing this will provide additional confidence in the results and allow us to better tune the evolution parameters so that solutions are found as efficiently as possible. In addition, better non-linear cost and mass functions will need to be integrated into the system.

Still this approach appears to offer human-competitive advantages by generating credible design concepts faster than humans are able to. Currently early mission concepts are created by a few experienced domain experts employing worst-case estimates. Automated design techniques provide anyone the ability to rapidly select and size design components, without bias, based on the space system's anticipated performance in the simulated environment.

ACKNOWLEDGEMENTS

The work described in this publication was carried out at the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration.

REFERENCES

[1] Terrile, R. J., Adami, C., Aghazarian, H., Chau, S. N., Dang, V. T., Ferguson, M. I., Fink, W., Huntsberger, T. L., Klimeck, G., Kordon, M. A., Lee, S., von Allmen, P. A. and Xu, J. (2005) "Evolutionary Computation Technologies for Space Systems" IEEE Aerospace Conference Proceedings, Big Sky, MT, March 2005.

[2] Robert Shishko, Robert G. Chamberlain, *NASA Systems Engineering Handbook*, NASA Publication SP-6105, Jet Propulsion Laboratory, 4800 Oak Grove Drive, Pasadena, CA 91109, second printing April 1996.

[3] Kalyanmoy, D., *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Ltd., 2001, p. 2, 172, 239, 173,

[4] Wertz, J.R., and W.J. Larson D.B., *Space Mission Analysis and Design*, Microcosm Press, 1999, p. 11

[5] Kordon, M., and E. Wood, "Multi-Mission Space Vehicle Subsystem Analysis Tools," IEEE Aerospace Conference Proceedings, Big Sky, MT., March 2003.

[6] Holland, J., *Adaption in Natural and Artificial Systems*. 1975, Ann Arbor: The University of Michigan Press.

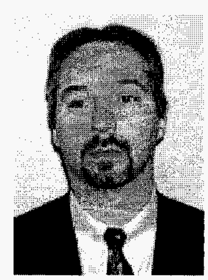
[7] Goldberg, D.E., *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison Wesley Longman, Inc., 1989.

[8] Daly, M., and M. Wilson, *Sex, Evolution and Behavior*, Second Edition. Wadsworth Publishing Company, Inc., 1978

[9] Levine, D., *PGAPack Parallel Genetic Algorithm Library*. Techn. Rep. Argonne Natl. Lab, ANL-95/18, 1995.

[10] Oberto, R., ,Team-X: The Advanced Spacecraft Conceptual Design Process, JPL Presentation, May. 2002

Richard J. Terrile *created and directs the Evolutionary Computation and Automated Design Center at NASA's Jet Propulsion Laboratory. His group has developed genetic algorithm based tools to improve on human design of space systems and has demonstrated that computer aided design tools can also be used for automated innovation and design of complex systems. He is an astronomer, the Mars Sample Return Study Scientist, the JIMO Deputy Project Scientist and the co-discoverer of the Beta Pictoris circumstellar disk. Dr. Terrile has B.S. degrees in Physics and Astronomy from the State University of New York at Stony Brook and an M.S. and a Ph.D. in Planetary Science from the California Institute of Technology in 1978.*



Mark Kordon *is the Technical Group Supervisor for the Modeling and Simulation Technologies Group, and Task Manager for Multi-Mission Analysis Tools at the Jet Propulsion Laboratory. His research interests include modeling and simulation techniques, evolutionary computing, multi-agent systems and space systems. Mark conceived and coordinated the work described in this paper. He received his Bachelor of Science in Computer and Systems Engineering from Rensselaer Polytechnic Institute.*

Dan Mandutianu *is a senior software engineer with the Modeling and Simulation Technologies Group. His research interests include distributed computing and synthesis of intelligent behavior in populations of simple agents. He received his MS in Computer Science and PhD in Electrical Engineering from the "Politehnica" University of Bucharest, Romania. Dan implemented the system architecture and reported the results described in this paper.*

Jose Salcedo *is a member of the technical staff in the Modeling and Simulation Technologies Group at the Jet Propulsion Laboratory. Jose received a Bachelor's of Arts in Physics from Occidental College('82) and a Master's Degree in Computer Engineering from the University of Southern California ('84). Jose implemented the fitness functions described in this paper.*

Eric Wood *is the lead developer of the Multi-Mission Power Analysis Tool at the Jet Propulsion Laboratory. He received his Bachelor of Science in Computer Science from California Polytechnic State University at San Luis Obispo. Eric provided MER and DI power subsystem data and flight rules used in this paper.*

Mona Hashemi is a member *of the technical staff in the Modeling and Simulation Technologies Group at the Jet Propulsion Laboratory. Mona received a Bachelors degree in computer science from California Polytechnic State University at Pomona 2003.*

Automated Design of Spacecraft Systems Power Subsystems

Richard J. Terrile, Mark Kordon, Dan Mandutianu, Jose Salcedo, Eric Wood and Mona Hashemi
Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive
Pasadena, CA 91109
818-354-6158
rich.terrile@jpl.nasa.gov

Abstract—This paper[1,2] discusses the application of evolutionary computing to a dynamic space vehicle power subsystem resource and performance simulation in a parallel processing environment. Our objective is to demonstrate the feasibility, application and advantage of using evolutionary computation techniques for the early design search and optimization of space systems. With this approach, engineers specify several sets of conditional subsystem performance criteria to trade off subsystem goals of mass, cost, performance and risk. Once specified, the integrated evolutionary/simulation software will then automatically generate a design option for each criteria, selecting and sizing power elements based on the space system's anticipated performance in the simulated environment. Initial Activity plans from two actual JPL missions, Mars Exploration Rovers (MER) and Deep Impact (DI) are used to test the software. Our results have shown human-competitive advantages by generating credible design concepts much faster than humans are able to and without the need for expert initial designs.

TABLE OF CONTENTS

1. INTRODUCTION

This work describes the application of evolutionary computational techniques to the automatic optimization of spacecraft power sub-systems. Over the past three years the Evolutionary Computation Group at NASA's Jet Propulsion Laboratory (JPL) has had the objective of demonstrating the feasibility, application and advantage of using biologically inspired evolutionary computational techniques for the early design search and optimization of space systems. In general, we have demonstrated that the same computational tools used for computer aided design and for design evaluation can also be used for the automated optimization of designs [1]. These multi-parameter design simulators are run on cluster computers as a parallel population of designs with randomly varying input parameters and starting with a random initial designs. The results are competed and selected down to a smaller sub-set of parents that provide the basis (using genetic operators of mutation and gene cross-over) for the design parameters of the next generation. Given a large enough population, sufficient generations and the right conditions for evolution, we have demonstrated the feasibility of automatically optimizing simulated designs.

We applied these evolutionary techniques by incorporating the Multi-Mission Power Analysis Tool (MMPAT) into an evolutionary framework running in a parallel processing environment. This tool is a dynamic space vehicle power sub-system resource and performance simulation, and is one of several multi-mission design tools in use at JPL. These tools use the spacecraft activity plan to simulate uplinked commands over the mission duration. With this approach, engineers specify several sets of conditional sub-system performance criteria to trade off sub-system goals of mass, cost, performance and risk. Once specified, the integrated evolutionary/simulation software will then automatically generate a design option for each criteria, selecting and sizing power elements based on the space system's anticipated performance in the simulated environment.

In order to quantify the advantage of these techniques we compared our automatically generated power sub-system designs to two actual JPL spacecraft designs. Theses tests were run using activity plans from the Mars Exploration Rovers (MER) and Deep Impact (DI) missions. The MER activity plan is for a landed mission spending 90 sols (Martian days) on the surface. Deep Impact is a comet flyby spacecraft with an 8.3 month-long activity plan that includes cruise from 1.0 to 1.5 AU from the sun. Initial populations were created from randomly selected parameters and design requirements identical to MER and DI were specified.

2. DESIGN LIFE CYCLE

At the Jet Propulsion Laboratory (JPL) the life cycle of a deep space mission normally goes through six phases, each culminating with a review by project management and its funding agencies [2]:

1 ——————————

[1] 0-7803-9546-8/06/$20.00© 2006 IEEE
[2] IEEEAC paper #528, Version 1, Updated Oct, 14 2005

- **Pre-Phase A:** Advanced Studies
- **Phase A:** Mission & System Definition
- **Phase B:** Preliminary Design
- **Phase C:** Design & Build
- **Phase D:** Assembly Test & Launch Ops
- **Phase E:** Operations

The process starts with Pre-Phase A where the goals and objectives of the mission are defined and several plausible mission concepts are created. These early mission concepts will trade off various elements in the design so that project managers can choose between different alternatives for mass, cost, performance and risk. Here a trade study is a process for seeking one or more optimal solutions when there are multiple, often conflicting, objectives. An optimal solution in this case means that if one objective improves, other objectives are compromised or traded off. The classic example of this is in car buying. Buyers must make a decision between cost and comfort since the less expensive cars are inevitably less spacious. This hypothetical trade-off is shown in Figure 1. To make select the design that best satisfies their requirements, the buyer would want to consider solutions that are evenly distributed along the Pareto-optimal front.

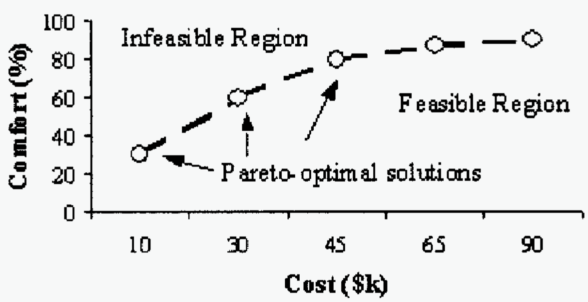


Figure 1 –Hypothetical Car Buying Trade-Off [3]

The cycle of goal definition, mission concept creation and design trade study is repeated many times in the early formulation phases. Each pass refines and improves the resolution of the design and removes design options from consideration. The product of this process is a single mission architecture characterized such that its effectiveness in achieving mission objectives can be properly evaluated. The mission architecture typically defines [4]:

- The Subject
- Orbit and Constellation
- Payload
- Flight System
- Launch Element
- Ground Element
- Mission Operations
- Command, Control and Communications Architecture

One important aspect of the mission architecture is the flight system. The purpose of the space vehicle flight system is to transport the payload safely to its destination and enable the return of science data to Earth. Typically the flight system is composed of several subsystems [2]:

- Power Subsystem
- Command & Data Handling Subsystem
- Telecommunications Subsystem
- Propulsion Subsystem
- Mechanical Subsystem
- Thermal Subsystem
- Guidance Navigation and Control Subsystem
- Spacecraft Flight Software

Each subsystem is responsible for a particular function, such as electrical power generation and distribution, and has design characteristics like solar array size, solar cell technology, secondary battery size and battery cell technology. Designing these subsystems to meet payload, trajectory, communication and activity requirements within the mass, cost and performance constraints of the project is vital for mission success. Automating this process would ensure consistent design quality while at the same time allowing experts to spend less time on routine tasks and more time evaluating various design options. This paper discusses an evolutionary computing approach for achieving the automated design of spacecraft systems.

With this methodology, evolutionary computing strategies are used with a dynamic, operations-validated power subsystem simulation to automate the design search and optimize space vehicle subsystem elements for a given set of project requirements and constraints. As we will demonstrate, this technique has several advantages over current approaches that rely on a small number of expert opinions employing worst-case estimates by generating credible power subsystem design concepts faster and for lower cost than humans are able to.

The paper begins with a brief overview of power subsystem design principles. It goes on to discuss the simulation used on this effort, the Multi-Mission Power Analysis Tool (MMPAT), and how it was integrated with an evolutionary computing framework. The paper continues by describing the fitness functions used on the effort and concludes by comparing the results generated to actual mission designs.

3. Power Subsystem Design

To properly develop a power subsystem fitness function it is important to understand the basic issues in spacecraft power subsystem design. Rather than consider all possible power sources it focuses on subsystems using photovoltaic solar power. Also, no consideration is given to cases where primary (non rechargeable) batteries might be used. After a brief discussion of some basic power subsystem design concepts, the various components are introduced in a logical order where later choices hinge upon the earlier selections.

Power Design Concepts

The main function of the electrical power subsystem is to generate and deliver electricity to all points of utilization on the spacecraft. The generated electricity must satisfy the power and energy requirements of the subsystems, be within the component ratings and voltage limits, as well as taking into account planned and unplanned usages of the spacecraft devices.

To handle interruptions in power generation, spacecraft power subsystems always provide an energy storage mechanism, such as a rechargeable secondary battery, for backup. To account for other unexpected problems, mission engineers establish flight rules to define the standard operating procedure for the spacecraft. Some of these rules add reserves such as power margin and energy margin to ensure reliable spacecraft operations. Others set critical thresholds such as minimum usable battery state of charge (SOC).

Power Margin

Power margin is the difference between the power generated and power consumed. When used as a flight rule, power margin establishes the minimum allowable power surplus at any point in time. This consideration is particularly important in cruise mode where it is necessary to keep the secondary battery fully charged in preparation for planned activities, solar obscurations or an unexpected contingency.

For example, when the Mars Exploration Rover (MER) was in cruise a flight rule was established requiring a power margin of 60 watts be maintained at all times, even during Trajectory Correction Maneuvers (TCM). This rule was more conservative than in other missions but was necessary since they used Lithium-Ion batteries that were untested in flight at that time.

Energy Margin

Energy margin is the difference between the energy generated and the energy used during a period of time. When used as a flight rule, energy margin establishes the minimum surplus of energy during a 24 hour period. On landed missions there are often periods where no power is generated, so stored energy from a secondary battery must be used to keep critical systems operational. In this case, the power subsystem design needs to ensure that a positive energy balance can be maintained while performing typical daily activities so that a low power condition does not arise that would interfere with activities or worse, put the mission at risk during the night.

For example, when the MER rovers were operating on the surface of Mars, an energy margin of 75 watt-hours was established by flight rule, so any plan created for the sol (Mars solar day) must store least this amount of energy during a full 24 hour sol. This includes battery charging and shunting, the total of which is defined as margin.

Usable State of Charge

A battery's available energy capacity, also known as state of charge (SOC), is measured as the amp-hours that the battery can deliver at the present discharge current before it reaches 100% depth of discharge. A battery is considered at a 100% depth of discharge when its cell voltage drops to a particular level. In spacecraft operations, the notion of SOC is modified to usable SOC so that engineers can monitor the energy available before the spacecraft begins to fail. Usable SOC measures the amp-hours that can be extracted from the battery before the bus voltage becomes less than the predefined minimum which would presumably cause the spacecraft to enter a low power state.

Usable SOC is computed by finding the "limiting SOC" that would necessarily exist at the present discharge current and the specified low bus voltage. The limiting SOC is then subtracted from the present SOC, giving the usable SOC.

Usable state of charge is useful when specifying a minimum allowable SOC in that it considers the capacity of the battery as well as the present rate of discharge. The MER flight rule used in cruise and surface specifies that usable SOC shall not be allowed below 5.5 amp-hours.

Energy Storage – Secondary Battery

On a spacecraft, energy is typically stored in the secondary battery. There are three battery design considerations: the battery chemistry, bus voltage range and battery size. Selection of the battery chemistry is based on several factors: mass/volume, charge efficiency, thermal characteristics, need for battery charger and lifetime. Presently there are two battery chemistries commonly in use: Lithium-Ion (Li-Ion) and Nickel-Hydrogen (NiH2). Table 1 compares the characteristics of these two battery chemistries.

In practice, the charge efficiency, thermal characteristics and small mass/volume favor Li-Ion for landed missions while the longer lifetime and the absence of a charger/cell balancer circuit favor using NiH2 for space missions. For example, MER used the Li-Ion battery chiefly due to its low mass and compact size. DI used the NiH2 battery since volume constraints were not an issue and at the time that the DI design was done there was no flight experience with Li-Ion batteries.

The bus voltage is selected to work well with the devices powered by the bus as well as being compatible with the battery chemistry selected. Typical bus voltage ranges are from 24 to 36 volts to conform to the NASA 28 Volt Bus design standard. Conformance to the standard provides a host of off-the-shelf devices and designs.

Table 1. Battery Chemistry Comparison Chart

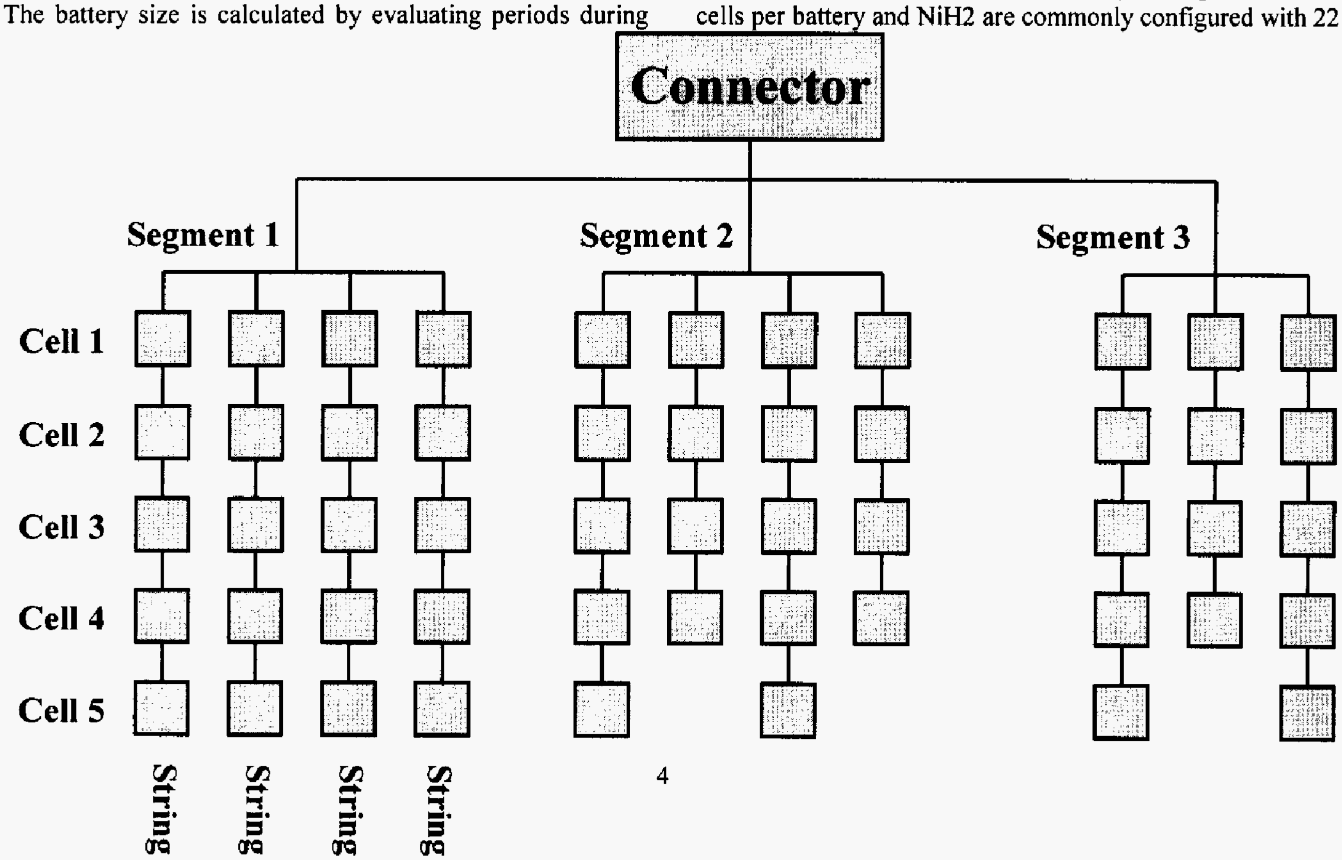
Battery Chemistry	Lithium-Ion	Nickel-Hydrogen
Mass efficiency	100 Watt-Hours/kg	30 – 60 Watt-Hours/kg
Volume efficiency	Relatively high	Relatively low
Charge efficiency	100%	Descends from 100% to 0% as SOC increases from ~70% to 100%
Thermal Characteristics	Moderate Thermal Load	Charge inefficiency (see above) becomes thermal load
Battery Charger/Cell Balancing Needed	Yes	No
Lifetime	2 - ?	Tens of years
Flight Experience	< 5	> 100

The bus voltage range and battery voltage are closely tied. Battery voltage is the product of cell voltage and the number of cells connected in series. Battery storage capacity is the product of cell capacity and the number of batteries connected in parallel. The bus voltage range and the number of battery cells in series should be selected so that at the maximum bus voltage the battery will attain a full or nearly full state of charge.

The battery size is calculated by evaluating periods during which loads exceed power generated (negative power balance). These periods can include night times for landed missions, solar obscurations for space missions as well as periods during which peak loads occur. The battery must be able to supply sufficient power during these periods to

Although batteries are custom made for each mission, in

practice Li-Ion batteries are commonly configured with 8 cells per battery and NiH2 are commonly configured with 22



cells per battery so that they operate well within the voltages dictated by the NASA 28 volt bus design.

Bus Voltage Control

As mentioned above, the bus voltage is selected to work well with the devices powered by the bus as well as being compatible with the battery chemistry selected. Typically, the bus voltage ranges are from 24 to 36 volts to conform to the NASA 28 Volt Bus design standard. The Bus Voltage Control (BVC) circuitry maintains the power bus within the specified bus voltage range. On spacecraft the bus voltage prevent the bus voltage from dropping below the minimum.control varies greatly but one of three approaches is often used: shunt limiter, string switching and BVC circuitry.

The shunt limiter approach restricts the power bus from exceeding the maximum allowed bus voltage by directing sufficient current to shunt resistors when the maximum bus voltage is reached. In this case all solar array current is accepted from the array with the excess being shunted.

String switching on the other hand, makes use of a string selection matrix that allows the BVC circuitry to effectively switch in and out individual strings of solar cells, thus limiting the amount of current received from the solar array. The BVC circuitry can allow for reducing the battery charge rate set point or bus voltage set point as the battery voltage and temperature increase. This is commonly associated with the NiH2 battery due to its tendency to heat up as its SOC nears full charge.

The Deep Impact mission used the string switching method due to its low mass made possible by the lack of shunt radiators. MER used the shunt limiter method because of the simplicity of the circuitry required and the small shunt radiators needed since it was operating in a cold gas environment.

Power Generation – Solar Array

Solar arrays are divided into segments. Each segment consists of a set of parallel strings of solar cells. Within a string, the solar cells are connected in series. The segments are in turn joined in parallel at the connector to the power bus. Figure 2 illustrates an example solar array configuration.

The voltage produced by a single string is the sum of the voltages produced by the cells in that string. The current produced by a string is equal to the current produced by the weakest cell. The current produced by a segment is the sum of the currents produced by each string in the segment. The voltage produced by a segment is equal to the voltage produced by the strongest string. The current produced by the array is the sum of the currents produced by each segment. The voltage produced by the array is the voltage of the strongest segment.

Figure 2. Example Solar Array Configuration

The voltage produced by the solar array must be sufficient to fully charge the battery, that is, it must exceed the maximum bus voltage to a degree that will provide a sufficient charge current. The current produced by the array must be great enough to recharge the battery to prepare it for periods when loads exceed array power production. Solar cell current rises in direct proportion to the amount of solar insolation striking it. Solar cell voltage drops as cell temperature increases.

The implication of the above two effects are that the closer to the sun a spacecraft is, the fewer strings it needs to produce the necessary current. However, at the same time more cells per string are needed to produce the necessary voltage.

On a spacecraft whose mission requires significant changes in the distance from the sun, this can lead to designs with some segments having more cells than other segments. The segments with more cells produce enough current when close to the sun, but as the spacecraft becomes further from the sun, the shorter strings begin producing sufficient voltage due to the lowered temperature and thereby begin contributing current, just when the current from the long strings becomes insufficient due to the lowered insolation.

The Deep Impact mission is a good example of these concepts. DI used two string lengths: 44 strings with 22 cells per string and 112 strings with 16 cells per string. The configuration provided adequate current and voltage when near earth and also when at encounter (1.5 AU) when the cells were colder (more voltage) and receiving less insolation (less current).

Figure 3 shows an example solar array IV curve when near Earth at a sun distance of 1.0 AU. Figure 4 shows the IV curve when it encounters the comet Tempel 1 at 1.5 AU. The solar array voltage required the battery is 35 volts. The current required by DI to provide adequate power margin after launch and during early cruise was expected to be 16 amps. The current required near encounter was expected to be 21 amps, due to greater heating requirements and the needs of the various instruments.

One can see by examining the 35 volt points that the required currents are provided in both cases. The near Earth case depends only on the longer strings while the encounter case requires both the long and short strings.

The solar array configuration on the MER rovers uses a mix of string lengths: 15, 16 or 17 cells per string. This is due both to restrictions on the area available for solar cells and the existence of shadow-casting masts. A shadow on a cell effectively removes that cell's contribution to the voltage generated by the string. Without sufficient voltage, a string cannot generate current. The number of unshadowed cells needed in a string was 15. The extra cells in a string would allow it to contribute current even if one or two of its cells were in shadow.

4. Power Simulator

An important aspect of this approach is the use of a dynamic spacecraft power subsystem simulation that has been validated in actual mission operations. Using such a simulation ensures that our design search is based on the anticipated operation of the subsystem rather than human estimates. This section describes in detail the features we required, the simulator that was selected and how the simulator's results maps to mission flight rules. MER and DI design parameters are listed in Table 2

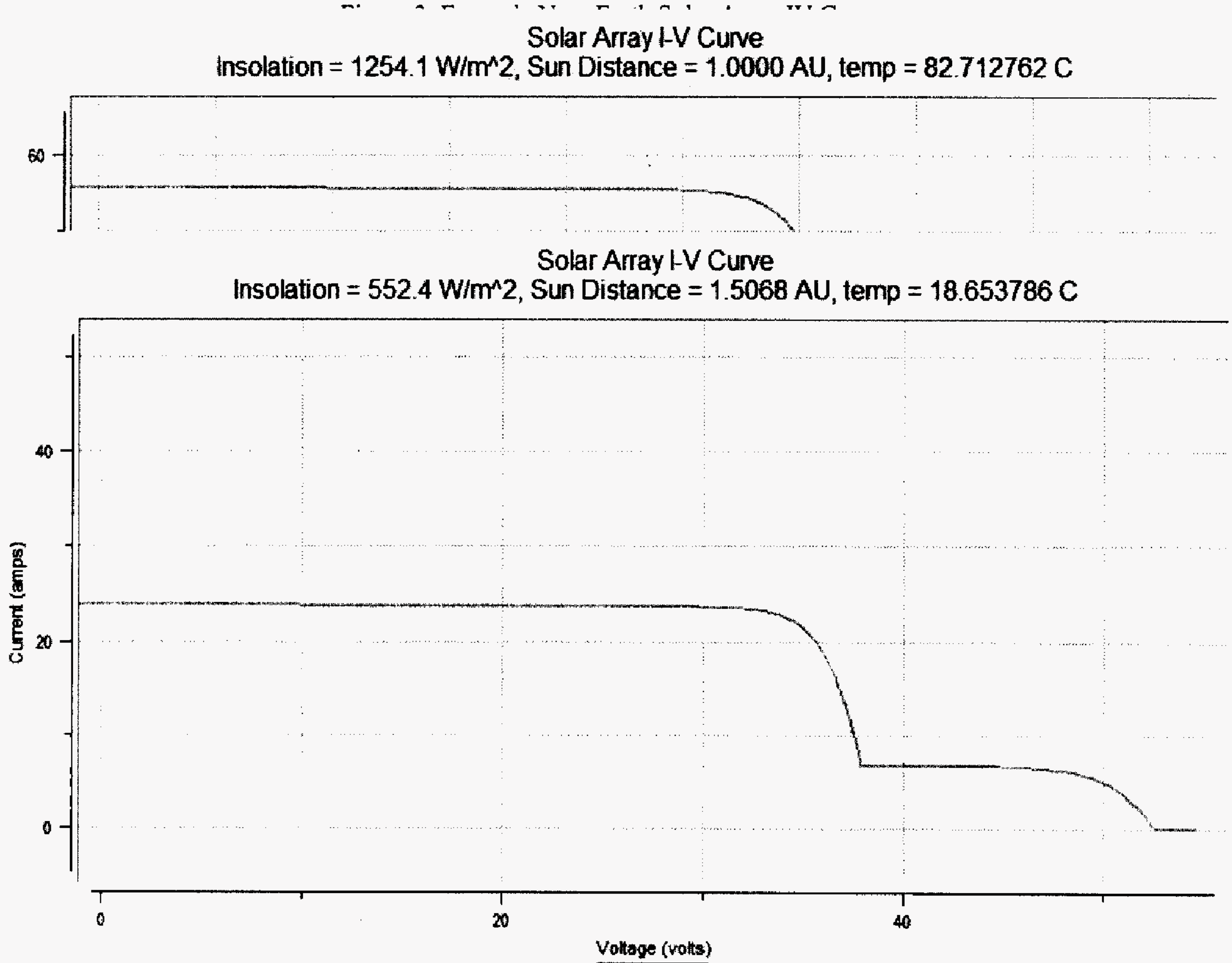


Figure 4. Example Encounter Solar Array IV Curve

Table 2 Summary of Deep Impact and MER Power Subsystem Flight Rules and Design Parameters

	MER Cruise	MER Surface	Deep Impact Cruise
Power Margin	60 watts	n/a	60 watts
Energy Margin	n/a	75 watt-hrs	n/a
Usable SOC	5.5 amp-hrs	5.5 amp-hrs	n/a
Total Number of Solar Array Segments	5	6	2
Total Number of Solar Array Strings	77	30	158
Total Number of Solar Cells	1310	480	2792
Battery Technology	Li-Ion	Li-Ion	NiH2
Battery Nameplate Capacity	20.0 amp-hr	20.0 amp-hr	16 amp-hr
Bus Control Method	Shunt Limiter	Shunt Limiter	String Switching
Bus Voltage	24-32.8	24-32.8	26.8-35
Mass of Solar Array Cell	0.01 kg	0.01 kg	0.01 kg
Cost of Solar Array Cell	$0.832k	$0.832k	$0.832k
Mass of Battery Cell	0.0167 kg/watt-hr	0.0167 kg/watt-hr	0.0167 kg/watt-hr
Cost of Battery Cell	$6.75k/amp-hr	$6.75k/amp-hr	$15.9k/amp-hr

Power Simulation Tool requirements

The task as defined required a simulation that could seamlessly handle multiple mission design alternatives and phases, and that could be integrated with an optimizer in a parallel processing environment. More specifically, the simulation needed to be a multiplatform library deployment with all of its design characteristics and state variables parameterized, and accessible through an Application Programming Interface (API). The API would also need to allow the user to enter an activity plan and trajectory.

Moreover, the simulation would need to use actual flight project data to quickly predict the resources and performance of the subsystem over the mission timeline, and would need to run in a closed loop manner with environment models that were, preferably, already integrated. Lastly, while not specifically required for this task, we wanted the simulation to be able to respond dynamically to inputs from other subsystems for compatibility with future research efforts.

Power Simulation in Operations

Ideally our simulator would be validated in operations. These operations tools have some unique input and output considerations in that each simulation must be able to input the design of their subsystem as well as a time-ordered sequence of events known as an activity plan. This plan is generated daily to determine what the spacecraft is intended to do. From this plan, a sequence is built and uploaded as instructions to the spacecraft.

To ensure that activity plans do not over commit resources and jeopardize the mission, the simulators must be able to predict what the resources will be after running the plan and verify that they are within the flight rule margins.

For example on MER, every sol the rover's battery state of charge and other key pieces of telemetry were supplied to a power simulator used by the mission planners to predict power generated, loads and the battery state of charge. Using this tool, they would then develop a sequence of rover commands that would include as much science and other useful activity as possible while still maintaining the required energy margin.

On Deep Impact, power analysts ran a power simulation that considered all expected loads plus all thermostatically-controlled heaters before each TCM. Power generation was simulated considering the sun distance and range of sun angles to be encountered during the maneuver. Results of the simulation were examined to be sure that the specified power margin would be maintained throughout the maneuver

Multi-Mission Power Analysis Tool

Given these requirements and the fact that we wanted a proven operations tool, we choose to use the Multi-Mission Power Analysis Tool (MMPAT) used on MER and Deep Impact. MMPAT is one tool in a suite of Multi-Mission Subsystem Analysis Tools at JPL [5]. It is a multiplatform software simulator currently used in Mars Exploration Rover (MER) operations to predict the performance and resources of space vehicle electrical power subsystems before a sequence of activities is uploaded.

The simulation can provide variable fidelity and produces dynamic time and sequence dependent results rather than static point solutions. As such, it models the behavior of power sources and energy storage devices as they interact with the spacecraft loads and the environment over a mission timeline at a level of detail appropriate to each stage of the project lifecycle, which in MER's case, is operations. The models in MMPAT include:

- Solar Array Model
- Solar Array Thermal Model
- Orbital Mechanics
- Astrodynamics Model
- Pointing Model
- Atmospheric Model
- Secondary Battery Model
- Secondary Battery/Thermostatically Controlled Heater Thermal Model
- Power Bus Model
- RTG Model
- Power Equipment List Model

All of the models were developed by power subsystem experts or adapted from validated heritage models. The tool itself comes with models for many of the most commonly used power sources, storage devices and power bus control methods used on space vehicles today. All of these models have been validated on previous or current missions, such as Pathfinder and MER, and give an accurate prediction of the system performance and resources.

The simulation is controlled by model parameters and was designed to be data-driven, modular and multiplatform. This means the models can be expanded to include additional hardware types. It also means that the application can be deployed stand-alone or as a library in another application, which in our case means integrated with an optimizer in a parallel processing environment. Moreover, the parameterized interface on MMPAT can also be used to change the mission type and analyze different mission phases since the tool supports the analysis of planetary landers, planetary orbiters, heliocentric orbiters and rovers as well as cruise, landed and orbiting phases and special events like flyby, TCM and EDL.

MMPAT Flight Rule Outputs

In some case MMPAT's outputs map directly to the flight rules, in other cases they do not. The usable SOC maps directly as the MMPAT variable: batt_usable_amphrs. However, while the MMPAT variable Emargin measures the energy margin of the spacecraft in watt-hours it accumulates the value over the lifetime of the mission. In ordered to compare this value to the daily energy margin flight rule, we needed to subtract the energy margin at the beginning of the day from the end-of-day value.

For power margin, MMPAT calculates the power margin in watts based on power actually taken off the solar array. Normally this would be fine except in the string switching algorithm, strings can be switched off. For a better comparison to the flight rule we want to use the power that could be generated from the solar array. The equation in this case was:

power margin = sa_Pavail - sa_Pactual + Pmargin

where:

sa_Pavail = Power available on array
sa_Pactual = Power being delivered by array
Pmargin = margin computed by MMPAT

5. SYSTEM ARCHITECTURE

With the flight rules defined and a power subsystem simulator selected, we now needed an automated method to search through the design space for different sizes and combinations of power equipment. We also need an architecture that allowed us to test the performance of each design using the simulation. The main architectural driver was the choice of optimization strategy. Since we wanted to find multiple optimal solutions in a single run, evolutionary computing was the natural choice. This section describes the evolutionary computing strategy and the resulting system architecture.

The Evolutionary Computing Strategy

Evolutionary computing seeks to generate optimal or near-optimal solutions for a given system by using a computer program to simulate the biological processes of natural selection [6,7]. This means that by using a process of random variation and selection through competition in an environment, the quality of solutions will iteratively improve. Simply put, the process involves generating a population of candidate solutions, evaluating how well they satisfy the requirements and constraints, and then randomly mating the solutions to create children for the next generation. The selection of mates is weighted toward the better solutions so that they will have a reproductive advantage. Implicit in this process is the notion of a particulate mechanism of inheritance.

In biology, organisms have a genetic coding known to as a genotype. Their morphology, physiology and behavior are referred to as the phenotype [8]. They are related to each other in that an organism's genotype describes influences and controls its phenotype. This means that changing an organism's genes will change its function, structure or behavior, and will oftentimes affect several characteristics at once since genes are typically pleiotropic. So in our application the design parameters are the genotype of the system, which succinctly describe and influence the structure and behavior of the subsystem or phenotype. Reproducing in this instance means contributing some design parameters from each parent to the child thus creating a combination of both of them that is hopefully better. This evolutionary process continues until some number of iterations has occurred or until the solution converges.

To support trade studies the system needed to be able to simultaneously generate multiple diverse solutions rather than a single point solution. This is achieved by allowing the user to create population slots that are used to bin segments of the population. Each slot is defined by a conditional that sets the membership criteria. The conditional is patterned after spacecraft flight rules for power margin, energy margin, usable SOC, and has the following form:

- Not less than x units

Where power margin units are in watts, energy margin units are in watt-hours and usable SOC is in amp-hours.

Typically cruise mission phases use power margin and landed missions use energy margin, but users can concatenate conditionals if desired. For example, a filter may be specified as 'not less than 20 watts and not less than 5.5 amp-hrs'.

The conditional may be modified by another conditional to bound the range of the slot. This optional conditional is in the form of:

- Not greater than x units

The conditional may also be modified by an exception that ignores the conditional for a certain period of time. The exception is in the form of:

- Except from t0 to t1

Exceptions may be concatenated. For example, to set a slot for a cruise mission with a trajectory correction maneuver that temporarily points its solar arrays away from the Sun the user could indicate 'not less than 20 watts except from day 10 to day 11 and not less than 5.5 amp-hrs'.

In this way the users may emphasize and any number of multiple solutions simultaneously.

PGASIM

The evolutionary mechanism described above is relatively straightforward to implement in software, but since there are numerous genetic algorithm frameworks on the market today we decided to use one that was already available. After a brief survey, we choose PGAPack developed by David Levine of the Mathematics and Computer Science Division at Argonne National Laboratory [9].

PGAPack is a general-purpose, data-structure-neutral, parallel genetic algorithm library. It is intended to provide most capabilities desired in a genetic algorithm library, in an integrated, seamless, and portable manner. The package consists of a set of library routines that supply the user multiple levels of control over the optimization process. The levels vary from default encodings, with simple initialization of parameters and single statement execution, to the ability to modify all relevant parameters in the optimization process at a low level. User written routines for evaluation or crossover and mutation can also be inserted if necessary.

Because the calculations of the fitness function involve computations that can be quite intensive, executing the evolutionary computing algorithm on massively parallel computers is essential for high-fidelity models. PGA Pack supports this by using the Message Passing Interface (MPI) for parallel execution on a number of processors. Thus the primary advantages that this package had over others is that it executed on cluster systems and is open source.

PGAPack did require some modifications for use in our application. Our changes to the package, which we are calling PGASIM to distinguish it from the original, were as follows:

1. Changed the code from C to C++.
2. The user provided functions are now methods in user defined classes.
3. A comprehensive configuration file is driving the algorithm instead of the user making explicit calls to set the parameters.
4. Increased flexibility by creating more place holders for custom code.
5. Enhance features like multi-criteria optimization and hybrid criteria.

Design Generation Process

At the beginning of an optimization run the head node reads the PGA and Optimization configuration files. This information is then used to instantiate the genetic algorithm data structures and create the initial population of power subsystem designs using randomly generated design parameters.

Once the initial population has been created, the compute nodes are configured by the head node to execute the simulation. Each compute node reads a reference MMPAT configuration and alters it according to the specific parameter values for that member. It also reads in the activity plan file, which is the same for all individuals, and executes the simulation.

After all of the members of a population have been simulated, the results are evaluated by the head node. First a check is made to determine if the design encountered any simulation errors that indicate if the design is infeasible, such as when the usable state of charge drops to or below zero. Then the output results from the simulation are examined and the member is put into a slot that satisfies the conditional. The ones that do not fit any slot are discarded. The members of each slot are then ranked from lowest to highest using the ranking criteria. Currently, there are two ranking criteria that can be specified by the user, either cost or mass.

The mass of the power subsystem is determined using a linear model. The mass of the power subsystem is obtained by determining the mass of the solar array and the mass of the battery. The solar array mass is determined by calculating the total number of solar cells and multiplying that value by mass-per-cell constant. Once this value has been calculated, the mass of the battery is determined by multiplying the battery's nameplate capacity by the mass-per-watt-hours constant. The sum of these two masses, determined the total mass of the power subsystem.

A similar model is used to determine the cost of the power subsystem. The cost of the solar array is calculated by multiplying the total number of cells with by a cost-per-cell constant; the cost of the battery, calculated by multiplying the battery's nameplate capacity by a cost-per-watt-hours constant.

When the entire population has been slotted and ranked, the head node collects the simulation results from the compute nodes and checks the stopping condition to decide whether the process should continue. If the stopping condition has been met, a report is generated containing the design parameters of the best population members. Otherwise the head node generates a new population.

The top members (lowest cost or mass) are selected to continue to the next generation so as not to lose the best candidates. The number of elites that are preserved for the next generation is calculated by subtracting the population replacement size from the population size. No duplicates are allowed in this elite survival list so if the same member exists at the top of another slot it is ignored and the next best candidate is selected. The rest of the replacement population is generated using a tournament operator to select parents and creating children by varying the parent's design parameters with crossover and mutation. After the new population has been generated, the cycle continues.

6. Test Cases

Two different mission scenarios were selected to test this automated design system, the MER surface phase and Deep Impact cruise phase. The intent was to test the system using different phases on different missions and compare the generated solutions to the actual mission design. Normally we would use a more idealized formulation phase activity plan for the optimization, but for a better comparison to the design, we used an actual activity plan from mission operations for both test cases. While this provided a good basis for comparison it also caused some problems because missions will occasionally violate their own flight rules. Because of this, we needed to mask out times when this occurred.

The optimization was run on two separate clusters to test system performance. The first is a Dell distributed memory parallel processor system with 3.2 GHz Intel Pentium 4 Xeon processors. The second is a Beowulf cluster with 450 MHz Intel processors. This section describes the tests cases and the results.

MER Surface Mission Test Case

MER Spirit Rover was used for the surface mission test case. The rover was placed in its actual location of 14.95 degrees south latitude and given an actual 90 Sol activity plan from mission operations. This corresponds to the original planned length of surface operations of MER-A at the Gusev Crater landing site. The design parameters that we were interested in generating and their MER values included:

Total Number of Solar Array Strings: 30
Total Number of Solar Cells: 501
Battery Technology: Li-Ion
Battery Nameplate Capacity: 20.0 amp-hrs
Bus Control Method: Shunt Limiter

For this optimization the following intervals were used for each of the variable design parameters:

Number of Strings : 1 to 40
Number of Cells per String : 1 to 40
Nameplate Capacity in amp: 1 to 30

The following cost and mass values were used and remained fixed throughout the analysis:

Solar cell cost: $0.832k
Solar cell mass: 0.01 kg per cell
Battery cost: $6.75k/amp-hr
Battery mass: 0.01667 kg/watt-hr

Using our system linear models to compute the total mass and cost of the Spirit configuration, we derived the following values:

Mass of the SA & Battery: 5.29 kg
Cost of the SA & Battery: $551.83k

Since this was a surface mission only the energy margin (EM) and usable SOC (USOC) flight rules were used. The following slots were defined as:

Slot 1: EM > 75 watt-hrs AND USOC > 5.5 amp-hrs
Slot 2: EM > 0 watt-hrs AND USOC > 5.5 amp-hrs
Slot 3: EM > 150 watt-hrs AND USOC > 5.5 amp-hrs
Slot 4: EM > 75 watt-hrs AND USOC > 10.0 amp-hrs
Slot 5: EM > 75 watt-hrs AND USOC > 1.0 amp-hrs
Slot 6: EM > 150 watt-hrs AND USOC > 10.0 amp-hrs
Slot 6: EM > 0 watt-hrs AND USOC > 1.0 amp-hrs

Time masks, in decimal hours, were defined to account for start up and late mission where flight rules were violated to add more science collection.

Surface Mission Results

Table 3 and 4 shows the designs generated on the Dell cluster after 200 and 400 generations, respectively, The first column shows the MER design parameters we were concerned with, while the remaining columns show the generated design parameters with slot 1 duplicating the MER power subsystem flight rules.

The first issue that needed to be addressed was the values of evolutionary parameters. Based on our experience we chose appropriate mutation and crossover probabilities on a population size of 128. Convergence of the design parameters using these evolutionary values occurred somewhere between 200 and 400 generations.

Table 3 MER Surface Mission Power Subsystem Designs after 200 Generations

	MER	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7
Energy Margin (watt-hrs)	> 75	> 75	> 0	> 150	> 75	> 75	> 150	> 0
Usable SOC (amp-hrs)	> 5.5	> 5.5	> 5.5	> 5.5	> 10.0	> 1.0	> 10.0	> 1.0
Total Number of Solar Array Strings	30	32	30	42	32	32	42	30
Total Number of Solar Cells	501	458	432	614	458	458	614	432
Battery Technology	Li_Ion	Li_Ion	Li_Ion	Li_Ion	Li_Ion	Li_Ion	Li_Ion	Li_Ion
Battery Nameplate Capacity (amp-hrs)	20.0	26.62	26.62	18.82	26.62	26.62	18.82	26.62
Mass of SA & Battery (kg)	5.28	5.02	4.76	6.45	5.02	5.02	6.45	4.76
Cost of SA & Battery ($k)	551.83	560.71	539.08	637.92	560.71	560.71	637.92	539.08

Table 4 MER Surface Mission Power Subsystem Designs after 400 Generations

	MER	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7
Energy Margin (watt-hrs)	> 75	> 75	> 0	> 150	> 75	> 75	> 150	> 0
Usable SOC (amp-hrs)	> 5.5	> 5.5	> 5.5	> 5.5	> 10.0	> 1.0	> 10.0	> 1.0
Total Number of Solar Array Strings	30	34	29	38	34	34	38	29
Total Number of Solar Cells	500	477	417	533	477	477	533	417
Battery Technology	Li_Ion	Li_Ion	Li_Ion	Li_Ion	Li_Ion	Li_Ion	Li_Ion	Li_Ion
Battery Nameplate Capacity (amp-hrs)	20.0	22.42	26.62	22.21	22.42	22.42	22.21	26.62
Mass of SA & Battery (kg)	5.28	5.14	4.61	5.70	5.14	5.14	5.70	4.61
Cost of SA & Battery ($k)	551.83	548.22	526.60	593.38	548.22	548.22	593.39	526.60

As can be seen, diverse solutions can be generated by varying the values in the flight rules. The slot with the lowest margin, and highest risk, generated the least expensive solution and the slot with the highest margin was the most expensive. Slot 1 where the MER flight rules were used generated design parameters that were less expensive and within 10% of the MER design. The exception was the design parameters for the number of strings which was larger than the MER design. This is probably because we did not use a flight rule or requirement to address the distribution of cells on the strings. Introducing a requirement for the voltage and/or current at the solar array connector would likely fix this problem.

Using 128 nodes, 400 generations of this 90 day activity plan took 7 hours to complete on the small clusters and less than one hour on the large cluster using 1000 nodes.

Deep Impact Cruise Mission Test Case

The Deep Impact mission was used for the cruise mission test case. The spacecraft trajectory starts from Earth or 1.0 Astronomical Units (AU) and travels an ellipse to 1.5 AU where it will encounter Comet Tempel 1. It took approximately 8.3 months to traverse this distance.

The activity plan was from DI operations. During the mission there were five trajectory correction maneuvers where the solar array edge-on to the Sun. This had the effect of forcing the battery to be the sole source of power to the spacecraft during this time.

The design parameters that we were interested in generating and their DI values included:

Total Number of Solar Array Strings: 158
Total Number of Solar Cells: 2792
Battery Technology: NiH2
Battery Nameplate Capacity: 16.00 amp-hrs.

For this optimization the following intervals were used for each of the variable design parameters:

Number of Strings : 1 to 40
Number of Cells per String : 1 to 40
Nameplate Capacity in amp: 1 to 30

The following cost and mass values were used and remained fixed throughout the analysis:

Solar cell cost: $0.832k
Solar cell mass: 0.01 kg per cell
Battery cost: $19.3k/amp-hr
Battery mass: 0.01667 kg/watt-hr

Using our system linear models to compute the total mass and cost of the DI configuration, we derived the following values:

Mass of the SA & Battery: 28.19 kg
Cost of the SA & Battery: $2361.82k

Table 5. Deep Impact Cruise Mission Power Subsystem Designs after 239 Generations

	DI	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
Power Margin	60	> 60	> 0	> 30	> 90	> 120	> 150
Total Number of Solar Array Strings	158	169	154	159	182	188	204
Total Number of Solar Cells	2792	2862	2676	2712	3218	3371	3830
Battery Technology	NiH2	NiH2	NiH2	NiH2	NiH2	NiH2	NiH2
Battery Nameplate Capacity (amp-hrs)	16.00	11.35	11.56	11.35	12.48	14.88	24.29
Mass of SA & Battery (kg)	28.19	28.81	26.95	27.31	32.39	33.96	38.71
Cost of SA & Battery ($k)	2577.94	2561.58	2410.23	2436.78	2875.87	3041.23	3572.84

Since this was a surface mission only the power margin flight rules was used. The following flight rules were used:

Power Margin: 60 watts

The following slots were defined:

Slot 1: Power Margin > 60 watts
Slot 2: Power Margin > 0 watts
Slot 3: Power Margin > 30 watts
Slot 4: Power Margin > 90 watts
Slot 5: Power Margin > 120 watts
Slot 6: Power Margin > 150 watts

Time masks, in decimal hours, were defined to account for trajectory correction maneuvers and the release of the impactor.

Cruise Mission Results

Table 5 shows the designs generated on the Dell cluster after 293 generations. The first column shows the DI design parameters we were concerned with, while the remaining columns show the generated design parameters. Slot 1 is identical to the DI power subsystem flight rules.

Using 51 nodes, 293 generations of this 8.3 month activity plan took 12 hours to complete on the small clusters and less than one hour on the large cluster using 1000 nodes..

7. Summary of Results

Results indicate that credible design concepts for spacecraft power sub-systems can be generated in an hour after about 20,000 MMPAT evaluations. Evolved designs showed slightly better cost and mass performance and were automatically generated as a trade study in a less than one hour. Multiple diverse designs with different mass, cost, performance and risk postures were generated providing project management and its funding agencies several plausible design concepts to choose from. These results demonstrate human-competitive advantages by generating credible design concepts faster that humans are able and without the need for initial design requiring domain expertise.

8. Conclusions

It takes a team of experienced JPL domain experts at least two weeks to generate a credible pre-award mission concept [10]. Results obtained using two different mission scenarios in different mission phases indicate that credible design concepts for a spacecraft power subsystem can be generated in an hour using evolutionary computing and a dynamic space vehicle power subsystem resource and performance simulation in a parallel processing environment. Moreover, multiple diverse designs with different mass, cost, performance and risk postures can be generated providing project management and its funding agencies several plausible design concepts to choose from. The results also seem to indicate that the speed of automated design generation appears to be driven primarily by the length of the mission to be evaluated and speed of the processing hardware, rather than the number of design parameters to be optimized.

Of course, several components are required to achieve good results:

1. System flight rules and/or technical requirements
2. Time-ordered list of S/C trajectory and state changes
3. A validated system simulation capable of ingesting time-ordered events, design parameters and outputting relevant system state variables
4. An optimization strategy
5. A parallel processing environment
6. A parallel- processing integration environment

Nevertheless, as promising as the results are there is still much work to be done before this system can be used in the flight project design process. On the implementation side, this approach also needs to be applied to other flight system subsystems. Once completed all of the subsystems need to be integrated together to provide more comprehensive design solutions.

In addition, it is likely that more power subsystem design parameters such as battery and solar cell technology will need to be optimized. It is important that when such variables are added, the simulation model supports the tradeoffs to be made and that a flight rule and/ or technical requirement is available that can be checked.

On the theoretical side, it will be important test the system against other missions. Doing this will provide additional confidence in the results and allow us to better tune the evolution parameters so that solutions are found as efficiently as possible. In addition, better non-linear cost and mass functions will need to be integrated into the system.

Still this approach appears to offer human-competitive advantages by generating credible design concepts faster than humans are able to. Currently early mission concepts are created by a few experienced domain experts employing worst-case estimates. Automated design techniques provide anyone the ability to rapidly select and size design components, without bias, based on the space system's anticipated performance in the simulated environment.

ACKNOWLEDGEMENTS

The work described in this publication was carried out at the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration.

REFERENCES

[1] Terrile, R. J., Adami, C., Aghazarian, H., Chau, S. N., Dang, V. T., Ferguson, M. I., Fink, W., Huntsberger, T. L., Klimeck, G., Kordon, M. A., Lee, S., von Allmen, P. A. and Xu, J. (2005) "Evolutionary Computation Technologies for Space Systems" IEEE Aerospace Conference Proceedings, Big Sky, MT, March 2005.

[2] Robert Shishko, Robert G. Chamberlain, *NASA Systems Engineering Handbook*, NASA Publication SP-6105, Jet Propulsion Laboratory, 4800 Oak Grove Drive, Pasadena, CA 91109, second printing April 1996.

[3] Kalyanmoy, D., *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Ltd., 2001, p. 2, 172, 239, 173,

[4] Wertz, J.R., and W.J. Larson D.B., *Space Mission Analysis and Design*, Microcosm Press, 1999, p. 11

[5] Kordon, M., and E. Wood, "Multi-Mission Space Vehicle Subsystem Analysis Tools," IEEE Aerospace Conference Proceedings, Big Sky, MT., March 2003.

[6] Holland, J., *Adaption in Natural and Artificial Systems*. 1975, Ann Arbor: The University of Michigan Press.

[7] Goldberg, D.E., *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison Wesley Longman, Inc., 1989.

[8] Daly, M., and M. Wilson, *Sex, Evolution and Behavior*, Second Edition. Wadsworth Publishing Company, Inc., 1978

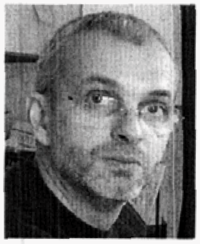
[9] Levine, D., *PGAPack Parallel Genetic Algorithm Library*. Techn. Rep. Argonne Natl. Lab, ANL-95/18, 1995.

[10] Oberto, R., ,Team-X: The Advanced Spacecraft Conceptual Design Process, JPL Presentation, May. 2002

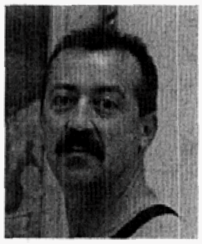
Richard J. Terrile *created and directs the Evolutionary Computation and Automated Design Center at NASA's Jet Propulsion Laboratory. His group has developed genetic algorithm based tools to improve on human design of space systems and has demonstrated that computer aided design tools can also be used for automated innovation and design of complex systems. He is an astronomer, the Mars Sample Return Study Scientist, the JIMO Deputy Project Scientist and the co-discoverer of the Beta Pictoris circumstellar disk. Dr. Terrile has B.S. degrees in Physics and Astronomy from the State University of New York at Stony Brook and an M.S. and a Ph.D. in Planetary Science from the California Institute of Technology in 1978.*

Mark Kordon *is the Technical Group Supervisor for the Modeling and Simulation Technologies Group, and Task Manager for Multi-Mission Analysis Tools at the Jet Propulsion Laboratory. His research interests*

include modeling and simulation techniques, evolutionary computing, multi-agent systems and space systems. Mark conceived and coordinated the work described in this paper. He received his Bachelor of Science in Computer and Systems Engineering from Rensselaer Polytechnic Institute.



Dan Mandutianu *is a senior software engineer with the Modeling and Simulation Technologies Group. His research interests include distributed computing and synthesis of intelligent behavior in populations of simple agents. He received his MS in Computer Science and PhD in Electrical Engineering from the "Politehnica" University of Bucharest, Romania. Dan implemented the system architecture and reported the results described in this paper.*



Jose Salcedo *is a member of the technical staff in the Modeling and Simulation Technologies Group at the Jet Propulsion Laboratory. Jose received a Bachelor's of Arts in Physics from Occidental College('82) and a Master's Degree in Computer Engineering from the University of Southern California ('84). Jose implemented the fitness functions described in this paper.*

Eric Wood *is the lead developer of the Multi-Mission Power Analysis Tool at the Jet Propulsion Laboratory. He received his Bachelor of Science in Computer Science from California Polytechnic State University at San Luis Obispo. Eric provided MER and DI power subsystem data and flight rules used in this paper.*

Mona Hashemi is a member *of the technical staff in the Modeling and Simulation Technologies Group at the Jet Propulsion Laboratory. Mona received a Bachelors degree in computer science from California Polytechnic State University at Pomona 2003.*