Optimization of Operations Resources via Discrete Event Simulation Modeling

B. Joshi[#], D. Morris*, N. White* and R. Unal[#]

[#] Dept. of Engineering Management
Old Dominion University
Norfolk, VA 23529

*Vehicle Analysis Branch
NASA Langely Research Center
Hampton, VA 23681

Presented at the 6th AIAA/USAF/NASA/ISSMO Symposium on
Multidisciplinary Analysis and Optimization
Bellevue, WA
September 4-6, 1996
AIAA 96-4181

Abstract

The resource levels required for operation and support of reusable launch vehicles are typically defined through discrete event simulation modeling. Minimizing these resources constitutes an optimization problem involving discrete variables and simulation. Conventional approaches to solve such optimization problems involving integer valued decision variables are the pattern search and statistical methods. However, in a simulation environment that is characterized by search spaces of unknown topology and stochastic measures, these optimization approaches often prove inadequate. In this paper, we have explored the applicability of genetic algorithms to the simulation domain. Genetic algorithms provide a robust search strategy that does not require continuity and differentiability of the problem domain. The genetic algorithm successfully minimized the operation and support activities for a space vehicle, through a discrete event simulation model. The practical issues associated with simulation optimization, such as stochastic variables and constraints, were also taken into consideration.

Introduction

The design of complex space systems, such as a launch vehicle, has historically emphasized the performance requirements aspect. However, declining budgets and global competition of recent years has prompted the need for economically competitive systems, designed for operability and supportability as well[21]. It has been shown that Operations and Support (O&S) activities account for 60% to 80% of life cycle costs of reusable space systems.[7,3] Therefore, in order to minimize life cycle costs, it is essential to study and optimize O&S resources in the early design phase.

The complex interaction between various O&S resources can be studied through a stochastic discrete event simulation model.[14,15] For a particular vehicle and space program, the model can be used to predict the number of missions flown and the mean vehicle turn-time, for user-specified resources. However, the simulation model cannot directly be used to minimize the O&S resources. Currently, this problem is addressed by manually varying the input variables of the simulation, one at a time, until the resource allocation seems to be at a minimum level. Minimizing the resources for a launch vehicle constitutes an optimization problem involving integer variables, which is a problem known to be “hard” to solve.[17] Furthermore, the simulation domain prevents explicit problem formulation in terms of objectives and constraints, thus rendering common mathematical programming methods inadequate.

Limitations of Conventional Optimization Methods

Gradient based optimization approaches, are the most common non-linear optimization techniques. However, gradient based methods cannot be applied under the conditions of discrete variables and discontinuities. Thus these cannot be used for problems such as the integer valued optimization problem outlined above. Methods that can be used to solve integer valued optimization problems include the pattern search and statistical methods. However, these techniques are not without certain drawbacks. The pattern search methods are local optima seeking methods.[9,16] However, in the simulation domain where the solution space typically is of unknown topology, these methods are susceptible to sub-optimization. Statistical methods, such as multiple comparison, and ranking and selection, yield the global optimum. However they are computationally intensive and their applicability in the simulation domain, where estimating a single stochastic measure or parameter may require several simulation runs, is severely restricted.[5] Therefore, these traditional approaches prove inadequate in the simulation domain.

Biased random search strategies such as genetic algorithms, have gained popularity as shortcomings of traditional optimization techniques have become apparent. Genetic algorithms have been applied to deterministic non-simulation environment with excellent results.[6] However, their applicability in the computationally intensive simulation domain remains to be explored. In this paper, we apply a genetic algorithm, to minimize the operation and support resources through simulation, as a step in this direction. The genetic algorithm is found to be a suitable and practical optimization technique that successfully minimized the maintenance resources.

The following sections contain a brief description of genetic algorithms. The specific operation and support optimization problem considered here is presented. Issues associated with stochastic simulation optimization, such as problem formulation and statistical accuracy, are also discussed. Finally, we present the results.

The Genetic Algorithm

Genetic Algorithms (GAs), first introduced by Holland are stochastic search algorithms inspired by the mechanics of natural selection and genetics[8]. GAs combine the principle of *survival of the fittest* among string structures with a structured yet randomized information exchange. They provide a robust alternative to traditional, numerical methods of optimization. Genetic algorithms have been proven, both theoretically and empirically, to provide a robust search in complex search spaces.[8,6]

The algorithm starts with a randomly selected initial population made up of encoded versions of candidate solutions to the optimization problem. Using the genetic operators of mutation and crossover, subsequent generations of strings are created from the current population. This cycle is repeated until a desired termination criterion is reached (for example, a predefined number of generations are processed).

Simple Genetic Algorithm:[19]

```
( initialize population;
 evaluate population;
 while termination criterion not reached
     ( select solutions for next populations
      perform crossover and mutation
      evaluate population ) )
```

Encoding Mechanism

Fundamental to the GA structure is the encoding mechanism for representing the optimization problem's variables. The encoding mechanism depends upon the nature of the problem variables. For continuous variables, the variable is first linearly mapped to an integer defined in a specified range. An integer is encoded using a fixed number of binary bits.

Fitness Function

Each string has a fitness function which represents a solution to the optimization problem. To maintain uniformity over various problem domains, a fitness function that normalizes the objective function between 0 and 1, is used. The normalized value of the objective function is the *fitness* of the string, which the selection mechanism uses to evaluate the strings of the population.

Selection Schemes

A selection scheme chooses the members of the population that will reproduce. A number of different selection schemes have been proposed over the years. In a simple genetic algorithm, a string with higher fitness function receives a higher number of offspring and has a higher chance of surviving in subsequent generations.

Crossover

Pairs of strings are picked at random from the population to be subjected to a single-point crossover. Assuming l is the string length, it randomly chooses a crossover point between 1 and l-1. An offspring is created by the portion of the first string up to the crossover point and the portion of the second string after the crossover point. After choosing a pair of strings the genetic algorithm invokes crossover only if a randomly generated number in the range 0 and 1 is greater than p_c, the crossover probability. The crossover probability is selected by the user.

Mutation

After crossover, strings are subjected to mutation. Mutation of a bit involves flipping it, i.e. changing a 0 to 1 and vice versa. The mutation rate p_m controls the probability that a bit will be flipped, and is set by the user. The bits of a string are mutated independently of one another.

Application to Operations and Support Modeling

A NASA model simulating the operation and support activities of launch vehicles during conceptual design, is used.[2] It uses estimated values for component reliability and maintainability, to simulate the mission, and pre- and postflight maintenance. Given the operation and support resources (crew and fleet size) the model predicts the number of missions flown and the mean vehicle turn-time for a particular space program. Underlying processes such as component and system failure, repair and replacement times, and maintenance delays are simulated. Due to the random nature of these processes, the simulation model and its outputs are stochastic.

Minimization of the operation and support resources requires the determination of the least fleet and maintenance crew size, that enables meeting the target mission rate in a timely manner. The decision variables to be minimized in this case, are restricted to integer values, within a certain ‘user-specified’ range. The constraints of meeting a given mission rate in a timely manner, are stochastic in nature, that are evaluated through simulation. This situation can be formally stated as a multiple objective optimization problem with integer variables. The objectives to be minimized are:

(1) Minimize the number of vehicles in the fleet.

(2) Minimize the crew size assigned to the power, structure, tanks, avionics, ther-

mal, auxiliary, life, mechanical and propulsion maintenance subsystem.

The stochastic constraints are:

(1) The mean of average delay should not exceed 48 hours
(2) The specified mission rate of 140 missions in five years, as denoted by the mean missions, is met.

Since the genetic algorithm handles a single objective, the various objectives (fleet size and nine different maintenance crew) to be minimized are combined additively through multi-attribute utility theory.[22] The additive utility function, recommended in the literature as a robust approximation when more than four objectives are involved is used.[11] Expert opinions from NASA engineers on life-cycle costs were used to develop relative utilities associated with each objective. The steps taken to assess the utility function were based on standard procedures, outlined in the literature[10]. This utility function, representing the fleet and manpower cost, forms a single objective function which is to be minimized. Symbolically the problem is stated as:

$$\text{Min } 100*v + c_1+c_2+c_3+c_4+c_5+c_6 + c_7+c_8+ c_9$$

subject to

$$E(delay) \leq 48 \text{ hours}$$
$$E(suc\_miss) \geq 140$$
$$1 \leq v \leq 5 \text{ positive integers}$$
$$1 \leq c_n \leq 40 \text{ positive integers}$$

where

v is the vehicles in the fleet,

c_n is the crew assigned to various maintenance subsystems,

E(delay), is a random variable representing the expected value of the average launch delay time over the specified time horizon, as predicted by the simulation model, and

E(suc_mis), is a random variable representing the expected value of the number of missions completed successfully, predicted by the simulation model.

Stochastic Simulation Optimization Framework

In the present example case, the average launch delay time and the missions flown constraints are stochastic and observed through simulation. This means that each replication of the simulation has the potential to give rise to a varying average delay and successful missions flown. Stochastic measures or constraints require special consideration during optimization. Standard statistical procedures recommended in the simulation literature are used to treat stochastic responses.[13,18,12,4]

Replications

One of the issues associated with a stochastic simulation is the number of replications or runs required in a study. This depends on the level of accuracy desired for estimating the average stochastic measures. The accuracy desired was specified in terms of confidence intervals by NASA engineers. For the present example, these were stated as follows:

	Desired width ±*w*	Desired conf. $(1-\alpha)$
Delay	±48 hours	80%
Successful Missions	±2 missions	95%

This implies that the mean of the average delay is to be estimated within ±2 days of the true mean, with an accuracy or confidence of 80%. Similarly, the mean successful missions flown are to be estimated within ±2 missions, with a 95% confidence.

The number of replications are computed based on the specified confidence level $(1-\alpha)$ of the true mean being within an interval $\pm w$ of the estimated mean. Representative distributions for the two stochastic measures were obtained by running the simulation 1000 times at a particular case. These were used to compute the number of replications in the standard manner[13,12]: The computation yielded the following replication requirements:

Delay:	19 replications.
Successful missions:	18 replications.

Hence, a conservative sample size of 20 was selected.

Chance Constraints

The variability inherent in stochastic constraints complicates the simulation optimization problem by forming fuzzy boundaries for the feasible region. This presents a danger of erroneously accepting a solution as feasible while it may have a high probability of being infeasible, and vice versa. The chance constraint theory approach has been used to convert the stochastic constraints into deterministic constraints. Chance constraints permit constraint violation up to a pre-specified probability limit.[1] The decision maker expresses a risk tolerance, in terms of a permissible probability of constraint violation. These were expressed as follows:

5% risk that mean of average delay exceeds 2 days.
5% risk that mean missions lag target of 140.

The above risk tolerances can be stated as:

$$P[E(delay) > 48\ hours] \leq 0.05$$
$$P[E(suc\_\ mis) < 140\ ] \leq 0.05$$

The chance constraints were implemented through statistical interval estimates for a pre-specified confidence, where the confidence denotes the probability that the interval contains the true parameter[20]. The limits at the specified confidence (5% risk or 95% confidence) provide deterministic boundaries for the infeasible region.

$$delay_{upp\_lim,\ .05} \leq 48\ hours$$
$$suc\_\ mis_{low\ \_lim,\ .05} \geq 140$$

The confidence intervals are estimated by using the Student's *t* distribution in the standard manner.[13]

Penalty Functions

Since a genetic algorithm does not allow for explicit constraint formulation, the objective function is penalized for constraint violation. The penalty is usually selected through a process of trial and error. Too large a penalty, may potentially neglect the original cost or objective function. On the other hand, a small penalty, may potentially ignore the effects of constraint violation. For the current example the following penalty functions determined experimentally, through a process of trial and error, were added to the objective function:

$$1000*(48\ hours - delay_{upp\_lim,\ .05})^2$$
$$1000*(140 - suc\_mis_{\ low\_lim,\ .05})^2$$

Results

The problem formulated above with multi-attribute additive function, chance constraints, and penalty functions was solved by applying a genetic algorithm. The genetic algorithm was run several times with different random number generator seeds. This practice of replication promotes genetic diversity in the population by sampling the search space more effectively. The algorithm was run for 20 generations at each replication. The algorithm parameters were determined through a process of trial and error as follows:

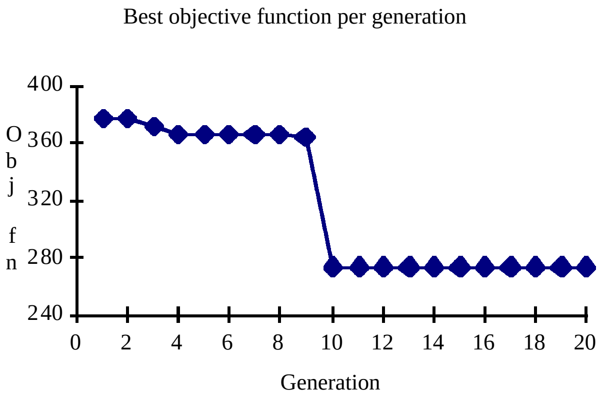
Population size:	50
Crossover probability:	0.9
Mutation rate:	0.2

The best objective function found among the GA replications was 273. This represents a 23% improvement in terms of utility, over the previous solution found by the 'one-variable-at-a-time' optimization approach[2]. The decision variables at this cost or utility of 273 are:

Vehicles: 2	Total crew: 73
Maintenance Subsystem	*Crew*
Power	6
Structure	7
Tanks	10
Avionics	6
Thermal	15
Auxiliary	7
Life	6
Mechanical	8
Propulsion	8

The stochastic constraints at this input combination were satisfied and well within their tolerance levels. The mean of average delay was 15.73 hours, with a 95% confidence that the delay does not exceed 38 hours. The target mission rate of 140 mission in a five year time span was achieved with a 95% confidence. The above stochastic measures (means of average delay and missions) were estimated at an 80% confidence of being within ±2 days, and 95% confidence of being within ±2 missions, of the true means respectively.

The following figure tracks the best objective function value over subsequent generations of the genetic algorithm. It can be seen that there is a gradual improvement in the cost function from the first generation to the ninth generation. The cost function at this point is attributed to three vehicles in the fleet. In the tenth generation, a sharp reduction in the cost function is visible. This corresponds to a feasible solution with two vehicles (273). There is no further improvement in the solution as the genetic algorithm progresses and it is said to have 'converged'.



Conclusions

Optimization of operation and support resources via discrete event simulation modeling is an integer programming problem known to be difficult to solve. The common approaches to optimizing integer decision variables, such as pattern search and statistical methods prove inadequate in the simulation domain. In this paper, we have examined the applicability of an emerging optimization technique to the simulation environment — the genetic algorithm. The genetic algorithm successfully minimized the operation and support resources through a simulation model. The solution produced in this manner resulted in a 23% improvement over the previous 'one-variable-at-a-time' approach. Furthermore, by incorporating statistical aspects in the optimization framework, a confidence level is associated with the solution itself, and the stochastic measures. Lastly, the applicability of the genetic algorithm in the simulation domain has been demonstrated.

Acknowledgment:

The genetic algorithm used here was developed by P. Gage of Stanford University.

References

1. Charnes A. and W. W. Cooper, "Chance Constrained Programming" *Management Science*, Vol. 6, No. 1, 1959.
2. Ebeling, C. E., and C.B. Donohue, Integrating O&S Models During Conceptual Design - Part III, Annual Report for NASA Langley, Grant No. NAG1-1-1327, 1994.
3. Fabrycky, W.J. and B.S. Blanchard, Life Cycle Cost and Economic Analysis, Prentice Hall, New Jersey, 1991.
4. Fishman, G.S., Principles of Discrete Event Simulation, John Wiley, New York, 1978.
5. Fu, M.C., "Optimization Via Simulation: A Review", *Annals of Operations Research*, Vol. 53, pp 199-247, 1994.
6. Goldberg, D., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company, Reading, 1989.
7. Griffin, J.J, "Whole Life Cost Studies: A Defense Management Perspective", *Engineering Costs and Production Economics*, No. 14, pp 107-115, 1988.
8. Holland, J.1975. Adaptation in Natural and Artificial Systems. The University of Michigan Press. Ann Arbor.
9. Hooke, R., and T.A. Jeeves, "A Direct Search Solution of Numerical and Statistical Problems", *Journal of Association for Computing Machinery*, Vol. 8, pp 212-229, 1961.
10. Huber, G. P., "Multi-Attribute Utility Models: A Review of Field and Field-Like Studies", *Management Science*, Vol. 20, No. 10, pp 1393-1402, 1974.
11. Keeney, R.L., and H. Raiffa, Decision with Multiple Objectives: Preferences and Value Tradeoffs, Wiley, New York, 1976.
12. Kleijnen, J., Statistical Tools for Simulation Practitioners, Marcel Dekker, New York, 1987.
13. Law, A., and W. D. Kelton, Simulation Modeling and Analysis, Mc-Graw Hill, New York, 1991.
14. Morris, W.E., T.A. Talay, and D.G. Eide, "Operations Simulation for the Design of a Future Space Transportation System", Proceedings of the AIAA 21st Aerospace Sciences Meeting, 1983.
15. Morris, W.D., N.H. White, W.T. Davis, and C.E. Ebeling, "Defining Support Requirements During Conceptual Design of Reusable Launch Vehicles", Proceedings of the AIAA 1995 Space Programs and Technologies Conference, 1995.
16. Nelder, J.A., and R. Mead, "A Simplex Method for Function Minimization", *Computer Journal*, Vol. 7, pp 308-313, 1965.
17. Papadimitriou, C.H., and K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Prentice-Hall, New Jersey, 1982.
18. Pritsker, A.A.B., Introduction to Simulation and SLAM II, Systems Publishing, West Lafayette, IN, 1984.
19. Srinivas, M and L. Patnaik, "Genetic Algorithms: A Survey", *Computer*, vol. 27, no.6, 17 pgs. 1994.
20. Teleb, R., and F. Azadivar, "A Methodology for solving multi-objective simulation-optimization problems", *European Journal of Operational Research*, Vol. 72, pp 135-145, 1994.
21. Unal, R., E. B. Dean and A. A. Moore, "Space Transportation System Operations and Support Cost Modelling Approach", *Journal of Parametrics*, Vol. X, No. 4, pp 35-51, 1990.
22. von Neumann, J., and O. Morgenstern, Theory of Games and Economic Behaviour, 2nd ed., Princeton University Press, Princeton, NJ, 1957.