JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 15, No. 3, May-June 1992

Decomposition Approach to Solving the All-Up Trajectory Optimization Problem

Todd J. Beltracchi*
Aerospace Corporation, Los Angeles, California 90009

This paper describes a new approach to solving the all-up (ground to mission) trajectory optimization problem. The problem considered in this paper is that of placing a maximum payload into a specified orbit, where a booster is used to place the payload in a low Earth parking orbit from which an upper stage transfers the payload to its mission orbit. One approach to solving the all-up problem has been to combine the booster and upper stage into one large simulation/optimization problem; however, this results in a difficult problem that is computationally expensive to solve. The algorithm proposed in this paper does not require any all-up trajectories to be explicitly optimized, but simulates separately the booster and upper stage. The algorithm is based on solving the maximum throw weight to a park orbit (for the booster), maximum payload transfer from the park orbit to the mission orbit (for the upper stage), and a coordination problem that adjusts the park orbit parameters to find the all-up optimum (maximum) payload to the mission orbit.

Nomenclature

A = matrix
$\boldsymbol{a}$ = vector, $(a_1, a_2, \ldots, a_n)$
$\boldsymbol{a}_B$ = quantities associated with the booster
$\boldsymbol{a}_s$ = scaled quantities
$\boldsymbol{a}_U$ = quantities associated with the upper stage
$\boldsymbol{D}$ = matrix of variable scale weights
$\boldsymbol{E}$ = matrix of constraint scale weights
f = objective function
g_i = ith inequality constraint
H = Hessian approximation for the system-level problem
HA = height of the apogee
HP = height of the perigee
h_i = ith equality constraint
L = Lagrangian function
$\boldsymbol{p}$ = vector of design parameters
Q_{max} = maximum dynamic pressure
RCS = reaction control system for upper-stage trim burns
RCS1 = propellant weight for first RCS burn
RCS2 = propellant weight for second RCS burn
$\boldsymbol{s}$ = search direction for the system-level problem
TW_B = throw weight of booster to park orbit
TW_U = transfer weight of upper stage
$\boldsymbol{u}$ = vector of Lagrange multipliers for inequality constraints
$\boldsymbol{v}$ = vector of Lagrange multipliers for equality constraints
$WP2$ = upper-stage propellent weight for SRM2
W_{SC} = spacecraft weight
$\boldsymbol{x}$ = vector of design variables
x_i = ith design variable
α = objective function scale weight
β = step length used in the line search
$\Delta \boldsymbol{p}_i$ = vector perturbation to the ith parameter, $[0, \ldots, \Delta p_i, \ldots, 0]$
ω_p = argument of perigee
$\nabla_x \boldsymbol{a}$ = gradient of a vector with respect to $\boldsymbol{x}$, $(\partial a/\partial x_1, \partial a/\partial x_2, \ldots, \partial a/\partial x_n)$

Superscript

$*$ = optimality

Presented as Paper 90-0469 at the 28th AIAA Aerospace Sciences Meeting, Reno, NV, Jan. 7-10, 1990; received July 24 1990; revision received March 19, 1991; accepted for publication March 27, 1991.

*Member Technical Staff, Optimization Techniques Section, Performance Analysis Department, P.O. Box 92957. Member AIAA.

Introduction

HIGH-fidelity simulation of trajectories for launch vehicles and spacecraft has become a common design task for aerospace enginers. Programs such as generalized trajectory simulation (GTS)[1] and program to optimize simulated trajectories (POST)[2] provide accurate simulation and optimization of vehicle trajectories. Numerically simulating trajectories can be an expensive task and optimization of the trajectories requires that many simulated trajectories be flown. Optimization procedures that use a minimum number of simulated trajectories to find the optimum are required.

The maximum throw weight of a variety of boosters (Titan, Delta, Atlas, STS) to several standard park orbits and the maximum orbit transfer capability (from these standard park orbits) for a variety of upper stages (IUS, PAM, Centaur) is routinely predicted. For some missions, the desired payload exceeds the capability of the booster/upper stage for standard park orbits. By flying the booster to a nonstandard park orbit, it is possible to obtain an increased payload capability. This is called an all-up optimization problem, and its solution is difficult to find. The high cost of placing payloads into near geostationary mission orbits dictates that the most efficient means of reaching the mission orbit should be used. By optimizing the all-up problem, extra payload can be delivered to the mission orbit. This can translate into extra spacecraft propellant which can be loaded, leading to an extended orbital lifetime.

A general systems engineering and integration task can require the capability to solve the all-up trajectory optimization problem. This requires accurate simulations of boosters and upper stages. Booster and upper-stage simulations can be combined to form the all-up problem, from which the all-up maximum payload capability of the booster/upper-stage combination is calculated. Because of the size and nonlinearity of the all-up problem, its solution can be expensive to obtain.

The objectives of this work are to reduce and simplify the computational effort required to solve the all-up trajectory optimization problem, and to predict if an all-up optimization will yield a significant increase (worth the effort of designing the new flight profile) in maximum payload capability over flying to a standard park orbit. The above issues are addressed by the proposed decomposition algorithm (which does not require any ground-to-mission orbit trajectories be explicitly simulated and optimized). The algorithm decomposes the all-up problem into solving for the maximum throw weight of the booster (to a park orbit), solving for the maximum orbital transfer weight of the upper stage (from the park orbit to the mission orbit), and a coordination (system integration) problem that determines the park orbit. The decomposition algorithm uses some of the concepts developed in Ref. 3.

Ideally, the decomposition algorithm should be coded as part of one large simulation; however, this is not necessary. The decomposition algorithm can be used, for example, when different people (or corporations) are running simulations of the booster and upper stage, to find the all-up optimum trajectory. The booster performance expert can coordinate with the upper-stage performance expert and use the decomposition algorithm to locate the all-up optimum park orbit. The decomposition algorithm also provides an estimate of how the all-up performance will increase when the booster and upper-stage trajectories are changed.

This paper is divided as follows: first, a standard form of the optimization problems is presented and the calculation of optimal parameter sensitivity derivatives is reviewed. Next, a description of the booster, upper-stage, and traditional all-up optimization problems is provided. Then the decomposition algorithm is described and the results for two test problems are presented. Finally, a section describing some future work is provided.

Standard Form

The standard form of the nonlinear programming problem (NLP) studied in this paper is

Minimize†

$$f(x,p) \tag{1}$$

Subject to

$$h_m(x,p) = 0, \qquad m = 1,\ldots,M \tag{2}$$

$$g_j(x,p) \geq 0, \qquad j = 1,\ldots,J \tag{3}$$

$$x = [x_1, x_2,\ldots,x_n]^T \tag{4}$$

$$p = [p_1,p_2,\ldots,p_k]^T \tag{5}$$

where f represents the objective function, a quantity such as payload, fuel margin, delta v, etc.; h represents equality constraints, quantities such as the final orbit elements equal to the mission orbit elements; and g represents inequality constraints, quantities such as current fuel load is less than or equal to the maximum fuel load. The design variables x are variables that the optimization algorithm is free to change. Typical design variables are pitch rates during the engine burns, amount of fuel loaded, coast times, and payload weight. The design parameters p are considered fixed during the optimization; however, these parameters may be subject to change. Typical parameters are dry weights of the stages, specific impulse of the engine, or the target orbit parameters. Optimum values are denoted by an asterisk.

†Maximation problems are problems that minimize $-f$.

The optimality criteria for the previous problem require

$$\nabla_x L(x^*,p) = 0,$$

$$h_m(x^*,p) = 0, \qquad m = 1,\ldots,M$$

$$g_j(x^*,p) \geq 0, \qquad j = 1,\ldots,J$$

$$u_j\, g_j(x^*,p) = 0, \qquad j = 1,\ldots,J$$

$$u_j \geq 0, \qquad j = 1,\ldots,J$$

where the following Lagrangian function is used

$$L = f(x,p) + \sum v_m\, h_m(x,p) - \sum u_j\, g_j(x,p)$$

where v_m and u_j are the Lagrange multipliers. Inequality constraints that are satisfied such that $g_j(x^*,p) = 0$ are called active constraints.

Sensitivity Analysis

Using a branch of nonlinear programming, called parameter sensitivity analysis, it is possible to predict how changes in the design parameters p affect the optimal design.[4] The generalized trajectory system (GTS) (simulation software) has a built in postoptimality/parameter sensitivity analysis routine.[5,6] Parameter sensitivity derivatives can be used to make linear predictions of how the optimum objective function changes when parameters are perturbed:

$$f^*(x^*,p + \Delta p_i) \approx f^*(x^*,p) + \frac{\mathrm{d}f^*}{\mathrm{d}p_i}\Delta p_i \tag{6}$$

where Δp_i represents a perturbation to the ith parameter ($\Delta p_i = [0,\ldots,\Delta p_i,\ldots,0]$), and $\mathrm{d}f^*/\mathrm{d}p_i$ is the sensitivity of the optimum value of the objective function with respect to the variations in the ith parameter. One way to calculate $\mathrm{d}f^*/\mathrm{d}p_i$ is

$$\frac{\mathrm{d}f^*}{\mathrm{d}p_i} = \frac{\partial f}{\partial p_i} + v^T\frac{\mathrm{d}h}{\mathrm{d}p_i} - u^T\frac{\mathrm{d}g}{\mathrm{d}p_i} \tag{7}$$

where v and u are the Lagrange multiplier vectors of the constraints. Predictions of how the optimum values of the design variables and Lagrange multipliers vary can also be made using parameter sensitivity derivatives.

If an equality constraint is written in the form

$$h_j(x,p) = h_j(x) - p_i = 0 \tag{8}$$

and p_i only appears in this one constraint, then the sensitivity of the objective function with respect to variations in p_i is equal to the Lagrange multiplier of the jth constraint. This result also holds for active inequality constraints of the above form.

Sensitivity derivatives can be used by mission planners to ask "what if" questions such as, "What is the new optimum payload weight if a different mission orbit is used?" or "What is the new optimum payload weight if the dry weight of stage II changes?" Parameter sensitivity derivatives provide good estimates of the optimum at perturbed values of the parameters when a reasonable size perturbation is made and there are no changes in the active set of constraints. Parameter sensitivity derivatives can also be used to obtain a prediction of the new location of the optimum. A more complete derivation of sensitivity results can be found in Refs. 4, 5, and 7.

The most efficient algorithms for solving general nonlinear programming problems are generally considered to be either generalized reduced gradient (GRG), generalized projected gradient (GPG), or recursive quadratic programming [(RQP) also known as sequential quadratic programming] type codes. The GRG,[8] GPG,[9] and RQP[8] methods automatically calculate estimates of the Lagrange multipliers, thus some parameter sensitivity information is obtained without extra computational effort. The efficiency of optimization codes is driven by problem nonlinearity, problem scaling, and the starting-point location. For trajectory optimization problems little can be done to correct for the built-in nonlinearity; however, the problems often can be "optimally" scaled, and due to the small domain of the problem a good initial guess of the optimum can often be made. Using sensitivity analysis, the starting point for perturbed cases can be estimated.

Problem Scaling

Problem scaling is considered an art in some circles because it is very difficult to properly scale a problem until the solution is known. If a problem is improperly scaled, this may result in either very slow convergence or nonconvergence. Problem scaling can be explained as numerically preconditioning the problem to transform real world quantitites into units that are amenable to computational manipulation. As an example, some constraints are expressed as radius: 1.0E + 6 ft = 0, velocity: 1.0E + 4 ft/s = 0, and flight-path angle = 0.

The fact that these constraints are of widely differing magnitudes can cause the optimization algorithm to converge slowly. The first two constraints can be scaled and are equivalent to

$$(\text{radius} - 1.0\text{E} + 6)/1.0\text{E} + 6 = 0$$

$$(\text{velocity} - 1.0\text{E} + 4)/1.0\text{E} + 4 = 0$$

The scaled constraints are usually much better behaved for the optimization, and thus it is recommended that some sort of scaling be performed before solving the problem.

The GTS system allows users to input scale factors for the objective function, design variables, and constraints. The scale factors are represented by

α = Objective function scale weight ($f_s = \alpha f$)

E = Diagonal matrix of constraint scale weights

$$\begin{pmatrix} h_s \\ g_s \end{pmatrix} = E \begin{pmatrix} h \\ g \end{pmatrix}$$

D = Diagonal matrix of design variable scale weights

$$(x_s = Dx)$$

The transformed problem (scaled) has the following properties:

$$\nabla_x f_s = \alpha D^{-1} \nabla_x f \tag{9}$$

$$\nabla_x \begin{pmatrix} h_s \\ g_s \end{pmatrix} = E \nabla_x \begin{pmatrix} h \\ g \end{pmatrix} D^{-1} \tag{10}$$

$$x_s = Dx \tag{11}$$

$$\begin{pmatrix} v \\ u \end{pmatrix} = \alpha^{-1} E \begin{pmatrix} v_s \\ u_s \end{pmatrix} \tag{12}$$

where the subscript s is used to denote a scaled quantity. The GTS system solves the scaled problem and unscaled results that are reported to the user.

The optimization algorithm used in this research, NLP2,[9] is sensitive to problem scaling. Hallman[10] has proposed an algorithm that optimally scales the problem at its solution. Scaling the problem so the active constraint Jacobian and projected Hessian are well-conditioned is usually the most beneficial form of scaling. Experience has shown that if a problem is optimally scaled then perturbed problems are often also well-scaled as long as the active set of constraints does not change. Again, problem scaling is esential for rapid convergence of trajectory optimization problems. It is also easier to meet a tight convergence criteria if a problem is properly scaled.

If a problem has not tightly converged, this can cause the sensitivity results to be inaccurate by affecting the accuracy of the Lagrange multiplier estimates.[6] It has been observed that if the Lagrange multipliers are calculated from the least-squares formulation

$$v, u = \min_{v,u} \| \nabla f + v^T \nabla h - u^T \nabla g \|_2 \tag{13}$$

this yields accurate estimates for well-scaled, tightly converged problems. However, due to numerical conditioning of the problem it has been observed that if the scaled Lagrange multipliers are calculated first by

$$v_s, u_s = \min_{v_s,u_s} \| \nabla f_s + v_s^T \nabla h_s - u_s^T \nabla g_s \|_2 \tag{14}$$

then transformed to the unscaled space via Eq. (12), the Lagrange multiplier estimates more closely resemble the true Lagrange multipliers for poorly scaled problems that did not converge tightly, where appropriate scale weights are chosen.

Description of the Boost, Upper-Stage, and All-Up Problems

This section describes the characteristics of the boost, upper-stage, and all-up optimization problems. The boost and upper-stage optimization problems are combined to form an all-up trajectory optimization problem. This requires accurate simulation of the booster and upper stage. The GTS system can accurately simulate a variety of boosters and upper stages and solve the all-up trajectory optimization problem. The GTS simulations used in this paper include J_2 oblate Earth effects and all orbital quantities referenced in this paper are osculating quantities (measured with respect to a spherical Earth model).

Boost Problem

The typical booster trajectory optimization problem is to maximize the booster throw weight (TW_B) to a given park orbit. Typical design variables are steering coefficients for the various stages, throttle settings, time of payload fairing jettison, payload weight, and so forth. Typically, there are constraints that the booster deliver the payload to a specified orbit (inclination, apogee, altitude, perigee altitude, argument of perigee), that maximum dynamic pressure be less than some design limit, and that the free molecular heating at payload fairing jettison be less than some preset limit.

Figure 1 represents a schematic of the booster that was used in this study. Some of the dry weights, propellant weights, and engine performance data are presented in Ref. 6. Constant

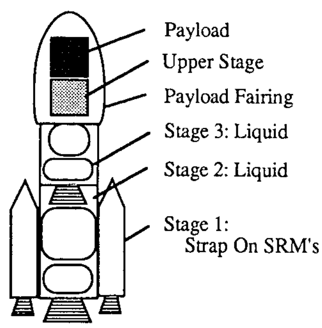

Fig. 1 Generic booster used to place upper stage and payload into park orbit.

thrust and weight flow were used for calculating the booster's engine performance. The booster consisted of two external strap on solid rocket motors (SRMs) and two liquid propulsion core stages. The booster problem is represented as

Maximize

$$TW_B \tag{15}$$

Subject to

$$\left.\begin{array}{l} FMH_{@\text{ fairing jettison}} \le \text{FMH}_{\text{spec}} \\ Q_{\max B} \le Q_{\max\text{ spec}} \\ |\alpha_{@\text{ SRM Jettison}}| \le \alpha_{\max} \end{array}\right\} \begin{array}{l}\text{Booster}\\ \text{constraints}\end{array} \tag{16}$$

$$\left.\begin{array}{l} HP_B = HP_{\text{Park}} \\ HA_B = HA_{\text{Park}} \\ \omega_{pB} = \omega_{p\text{Park}} \end{array}\right\} \begin{array}{l}\text{Park orbit}\\ \text{constraints}\end{array} \tag{17}$$

$$x = \left\{\begin{array}{l} \text{Pitch rates for booster stages} \\ \text{Time of fairing jettison} \\ TW_B \end{array}\right\} \tag{18}$$

$$p = \left\{\begin{array}{l} \text{Inert weight of stages} \\ \text{Weight of payload fairing} \\ \text{Park orbit definition} \end{array}\right\} \tag{19}$$

where TW_B is the booster throw weight, *FMH* is the free molecular heating, $Q_{\max}$ is the maximum dynamic pressure encountered, and α is the angle of attack. HP_{Park}, HA_{Park}, and $\omega_{p\text{Park}}$ are the height of perigee, apogee, and argument of perigee at park orbit insertion, respectively. The problems studied in this paper use two pitch rates for each stage of the booster, the time of payload fairing jettison, and the throw weight as design variables for the booster problem. The cases studied in this paper used a fixed launch azimuth (and launch pad location) for the booster, resulting in park orbits with a 28.622-deg inclination. The true anomaly at park orbit insertion was not constrained.

The trajectory simulation of the booster is propagated by numerically integrating the equations of motion. Because of the nonlinearity in the thrust, gravity, and aerodynamic forces, small time steps must be used to perform the integration. This makes the simulation of the boost phase computationally expensive. Often atmospheric, aerodynamic, and propulsive forces for the booster are represented by linear interpolation data tables; however, the simulation in this paper used the curve-fit approximations for the physical processes estimation, instead of using table data. Luke[11] reports that curve-fit approximations reduce the cost of optimizing the problem without sacrificing the accuracy of the solution.

Upper-Stage Problem

The typical trajectory optimization problem for an upper stage is to maximize the payload transferred from a park orbit to some mission orbit. For most orbit transfers, orbital mechanics dictates that either two or three burns are optimal. Typical design variables for an upper-stage optimization are steering coefficients (pitch and yaw) for the burns, amount of fuel used for each burn, coast times between the burns, and payload weight. The orbit at upper-stage burnout is constrained to be equal to the mission orbit, and the stack weight (weight of the upper stage plus the spacecraft) is constrained to be equal to the throw weight that the booster can place in the park orbit.

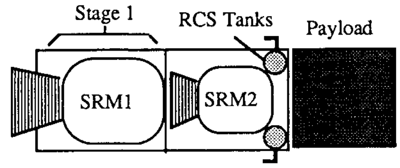

Fig. 2 Generic upper stage with SRMs.

The generic upper stage used in this study consists of two large SRMs denoted by SRM1 and SRM2. Additionally, after each SRM burn there is a small reaction control system (RCS) trim burn, which also is used to correct for dispersions in performance of the SRMs (Fig. 2). The upper-stage problem is represented by

Maximize

$$TW_U \text{ (payload weight)} \tag{20}$$

Subject to

$$\left.\begin{array}{l} \text{Radius} = 138{,}334{,}443 \text{ ft} \\ \text{Inertial velocity} = 10{,}087.4532, \text{ ft/s} \\ \gamma = 0 \text{ deg} \\ \text{Latitude} = 0 \text{ deg} \end{array}\right\} \begin{array}{l}\text{Mission}\\ \text{orbit}\\ \text{constraints}\end{array}$$

$$\text{Azimuth} = 90 \text{ deg} \tag{21}$$

$$\text{Payload weight} + \text{Initial upper-stage weight} = TW_B \tag{22}$$

$$\text{RCS1 (Vernier propellant weight)} = 35 \text{ lb} \tag{23}$$

$$\text{RCS2 (Vernier propellant weight)} = 115 \text{ lb} \tag{24}$$

$$\text{WP2 (Weight SRM2 propellant)} \le 6500 \text{ lb} \tag{25}$$

$$x = \left\{\begin{array}{l} \text{Pitch and yaw attitudes for SRMs} \\ \text{Delta } V \text{ for vernier burns} \\ \text{WP2} \\ \text{Payload Weight} \end{array}\right\} \tag{26}$$

$$p = \left\{\begin{array}{l} \text{Park orbit definition} \\ \text{Inert weights of stages} \end{array}\right\} \tag{27}$$

where TW_U is the payload weight delivered from the park orbit to the mission orbit. The mission orbit specified by Eq. (21) represents a geostationary mission orbit. The constraint [Eq. (22)] is used to assure that the booster can place the upper stage and payload into the park orbit. Constraints [Eqs. (23) and (24)] are used to allocate the fuel for the RCS trim burns. Approximately 100 lb of RCS propellant are held in reserve for attitude control and to correct for any dispersions in the SRM performance. The propellant weight for SRM1 was fixed and the propellant weight for SRM2 (WP2) was constrained [Eq. (25)] to be less than 6500 lb.

After an upper-stage trajectory optimization, a sensitivity analysis is performed to determine how design parameters affect the optimum. Typical design parameters for the upper-stage sensitivity analysis are parameters defining the park orbit perigee, apogee, argument of perigee, true anomaly, throw weight of the booster, dry weights of the stages, and specific impulse of the engines.

The performance of some upper stages can be estimated using impulsive burn approximations and the equations of motion for the coast periods can be analytically solved instead of having to use numerical integration of the equations of motion. The results generated in this study use a J_2 approximation for propagation of the coast periods.[12] If care is used setting up the impulsive burn approximations, then the solution closely matches the solution found using the more expensive numerical integration of the equations of motion. When the impulsive burn approximation is used the optimum trajectory for the upper stage can be found in much less CPU time than that required to solve the boost problem.

All-Up Problem

The problem of determining the optimum all-up trajectory involves linking together the simulation of the booster and upper stage and optimizing the total problem for maximum payload. As mentioned in the Introduction, this results in a larger, more difficult optimization problem. The all-up problem is represented by

Maximize

$$W_{SC} \text{ (Spacecraft weight)} \tag{28}$$

Subject to

$$\text{Mission orbit} = \text{mission orbit}_{\text{spec}} \text{ [Eq. (21)]} \tag{29}$$

$$\text{Booster constraints (Eq. 16)} \tag{30}$$

$$\text{Upper-stage constraints [Eqs. (23–25)]} \tag{31}$$

$$x = \begin{Bmatrix} \text{Booster variables [Eq. (18)]} \\ \text{Upper-stage variables [Eq. (26)]} \end{Bmatrix} \tag{32}$$

$$p = \begin{Bmatrix} \text{Park orbit definition} \\ \text{Inert weights of stages} \end{Bmatrix} \tag{33}$$

where W_{SC} is used to represent the spacecraft weight for the all-up problem [Eqs. (28–33)] as opposed to TW_U, which is used for the upper-stage transfer weight from the park orbit [Eqs. (20–27)]. The traditional all-up simulation is created by combining the booster and upper-stage simulations. The constraints include mission orbit, booster, and upper-stage constraints. In the all-up simulation, the park orbit and booster throw weight constraints are implicitly satisfied by initializing the upper stage at booster burnout. The design variables are a combination of the design variables for the booster upper-stage problems.

The all-up trajectory optimization problem is composed of approximately eight design variables for the boost phase and 12 design variables for the upper stage, for a total of 20 design variables. This problem may have eight or nine active constraints at the solution, leaving up to 12 degrees of freedom. The constraints for the all-up problem are very nonlinear; this adds to the complexity of solving the problem. Even the most powerful optimization algorithms require that many trajectories be simulated, and each trajectory evaluation is expensive because both the booster and the upper stage must be simulated.

Luke[13] reports that special procedures are required to efficiently solve the all-up optimization problem, and that considerable computational savings are possible if care is used in generating the starting point for the all-up problem. If an estimate of the optimal park orbit is available, then the starting point for the all-up problem can be calculated by solving a maximum performance booster problem to the park orbit, and solving for maximum upper-stage performance from the park orbit to the mission orbit.

For all-up problems there are some design variables that have very little effect on the all-up optimum spacecraft weight.[6] If these variables are left in the problem, they can cause it to be poorly scaled and the optimization algorithm requires extra iterations (and function evaluations) to find the optimum trajectory with only a very small increase in W_{SC}. If these design variables are held constant (declared as design parameters), the problem can be solved much more efficiently. A parameter sensitivity analysis should be performed on the design variables that were frozen as design parameters. This verifies that the estimated optimum value of the parameter is not too far from the true optimum value, and the maximum payload would not increase if the parameter were optimized.[6] Often these uninfluential (with respect to maximum spacecraft weight) variables are associated with the upper stage and are removed to simplify the problem and allow timely convergence of the optimization algorithm.

Decomposition Algorithm

Solving the all-up problem is computationally expensive if the optimization algorithm is not provided with a good initial guess and the problem is poorly scaled. This section proposes an algorithm that divides the design task between optimization of the booster and optimization of the upper-stage trajectories. A logical set of interface variables is used to decouple the problem, and parameter sensitivity derivatives are used to calculate how changes in the interface variables affect the all-up spacecraft weight. This section describes the algorithm and an illustrative example is used to demonstrate how the algorithm can be applied.

The decomposition algorithm described in this paper is different from that proposed by Petersen et al.[14] The work of Petersen broke the problem into a series of full-rank targeting subproblems and a master control problem. The algorithm described in this paper uses subproblems that are typical for performance analysis of boosters and upper stages but uses postoptimally parameter sensitivity analysis to calculate the sensitivity of changes in the park orbit. As in the Petersen work, this work maintains a series of feasible trajectories approaching the solution; however, in the combined formulation of the all-up problem [Eqs. (28–33)] a series of feasible trajectories can be maintained if a feasible point optimization algorithm is used.

The decomposition algorithm's purpose is to locate the optimal park orbit for the mission. In this development, the park orbit was specified by three orbital parameters: apogee, perigee, and argument of perigee. First, the maximum throw weight of the booster (TW_B) to a candidate park orbit is solved and the sensitivity of the throw weight, with respect to variations in the park orbit, is estimated. Then the maximum payload (TW_U) orbit transfer capability of the upper stage from the park orbit to the mission orbit (constraining the upper stage plus payload weight to be equal to the booster throw weight TW_B) is solved, and the sensitivity of the maximum payload (W_{SC}) with respect to park orbit parameters is calculated. By using the chain rule the sensitivity of the maximum payload to the mission orbit, with respect to the park orbit, is calculated and a new set of park orbit parameters is generated. This procedure is repeated until convergence occurs.

The basis of the decomposition algorithm is in writing the composite objective function for the system-level problem:

Maximize

$$W_{SC}[HP_{\text{Park}}, HA_{\text{Park}}, \omega_{p\text{Park}}, \nu_{\text{Park}}(\omega_{p\text{Park}}), TW_B(HP_{\text{Park}}, HA_{\text{Park}}, \omega_{p\text{Park}})] \tag{34}$$

This represents a composite function of the booster and upper-stage problems. If the launch azimuth of the booster is allowed to vary, then the inclination of the park orbit can be included in the argument list in Eq. (34). The chain rule is applied to Eq. (34) to obtain the derivatives for the system-level problem.

$$\frac{dW_{SC}}{dHA_{\text{Park}}} = \frac{\partial TW_U^*}{\partial HA_{\text{Park}}} + \frac{\partial TW_U^*}{\partial TW_B}\frac{\partial TW_B^*}{\partial HA_{\text{Park}}} \tag{35}$$

$$\frac{dW_{SC}}{dHP_{\text{Park}}} = \frac{\partial TW_U^*}{\partial HP_{\text{Park}}} + \frac{\partial TW_U^*}{\partial TW_B}\frac{\partial TW_B^*}{\partial HP_{\text{Park}}} \tag{36}$$

$$\frac{dW_{SC}}{d\omega_{p\text{Park}}} = \frac{\partial TW_U^*}{\partial \omega_{p\text{Park}}} + \frac{\partial TW_U^*}{\partial \nu_{\text{Park}}}\frac{\partial \nu_{\text{Park}}}{\partial \omega_{p\text{Park}}} + \frac{\partial TW_U^*}{\partial TW_B}\frac{\partial TW_B^*}{\partial \omega_{p\text{Park}}} \tag{37}$$

The terms in Eqs. (35–37) are found at the solution of the booster and upper-stage problems using in Eq. (7). The $\partial TW_B^*/\partial HA_{\text{Park}}$, $\partial TW_B^*/\partial HP_{\text{Park}}$, and $\partial TW_B^*/\partial \omega_{p\text{Park}}$ are available at the solution of the booster problem as the Lagrange multipliers of the park orbit constraints. $\partial TW_U^*/\partial TW_B$ in the upper-stage problem is the Lagrange multiplier of the throw weight constraint for the upper stage. The sensitivity of the upper stage, with respect to variations in park orbit, ($\partial TW_U^*/\partial HA_{\text{Park}}$, $\partial TW_U^*/\partial HP_{\text{Park}}$, $\partial TW_U^*/\partial \nu_{\text{Park}}$, and $\partial TW_U^*/\partial \omega_{p\text{Park}}$) are found using Eq. (7). The $\partial \nu_{\text{Park}}/\partial \omega_{p\text{Park}}$ is equal to -1 by definition of the true anomaly [ν(true anomaly) $= u$(argument of latitude) $- \omega_p$(argument of perigee)].

The derivatives in Eqs. (35–37) can be used to predict how the maximum payload changes if a nonstandard park orbit is used. This algorithm can be terminated before it converges and the best point can be used as a starting guess for the traditional all-up optimization problem [Eqs. (28–33)].

Since the upper-stage problem is decoupled from the booster problem, some upper-stage design variables that were not considered (because experience has shown they have a small effect on maximum all-up payload) as part of the all-up problem can be considered in the analysis of the upper stage in the decomposition algorithm. The solution of the upper-stage problem is cheap compared to the booster problems; thus the addition of the insensitive variables should have little effect on the overall computational cost and only cause a minimal increase in the cost of solving the upper-stage problem.

The booster and upper-stage problems have less than 5 degrees-of-freedom; thus, after the first booster and upper-stage problems were solved "optimal" scale factors were calculated. The scale factors were calculated by an algorithm that minimizes the condition number of the active constraint Jacobians and the projected Hessian.[10] These scale factors were used to precondition the problems for all subsequent iterations. The scale factors resulted in perturbed problems which were solved efficiently. The Lagrange multiplier estimates were calculated using the scaled gradients and then unscaled because this resulted in more accurate estimates of the sensitivity derivatives.

Variable metric methods can be used to accelerate the search for the point (HP_{Park}, HA_{Park}, $\omega_{p\text{Park}}$) where the gradient of the system-level problem [Eqs. (35–37)] is equal to zero indicating optimality. A new estimate of the optimal park orbit is calculated by solving

$$s = H^{-1}\nabla_{(HP_{\text{Park}}, HA_{\text{Park}}, \omega_{p\text{Park}})} W_{SC}$$

where H is an approximation to the Hessian of the system problem. The following line search is used to obtain a new estimate of the optimum park orbit.

$$(HP_{\text{Park}}, HA_{\text{Park}}, \omega_{p\text{Park}}) = (HP_{\text{Park}}, HA_{\text{Park}}, \omega_{p\text{Park}}) + \beta s$$

where β is often set equal to one. As the variable metric method converges, the optimal step length β approaches one.

NLP2,[9] a GPG algorithm, was used to solve the booster and upper-stage problems. The system problem was solved manually using the SR1/PD/BFS[15] variable metric update, with cubic interpolation, if steps of length one did not produce an improved objective value.

The system-level problem could have been solved using the BFGS update; however, this would require a more exact line search for convergence. Powell[16] has shown that the BFGS update can have trouble locating the solution of problems if inexact line searches are used. The best algorithm to solve the system-level problem is an area of active research.

Test Problems

The decomposition algorithm was applied to two test problems and the solution was compared to the solution obtained using the all-up approach [Eqs. (28–33)]. The mission orbits used are geostationary [Eq. (21)] and a 1-deg inclined geosynchronous, the perigee altitude of the park orbit was constrained to be 85 nm. The 1-deg inclined case is summarized; a more complete solution is available in Beltracchi.[17] The solutions obtained by the decomposition algorithm closely matched those obtained using the traditional approach.[6] No comparisons were made of the computation cost of the decomposition vs the traditional formulation in this initial study because the large overhead associated with solving the problems. The decomposition algorithm's performance can be improved if it can be automated, but this would require significant effort.

Case 1: All-Up Performance to a Geostationary Mission Orbit

The first iteration of the method for solving the all-up problem to a geostationary mission orbit is presented to illustrate how the method can be used to find the all-up maximum payload. The initial Hessian approximation for the system-level problem was taken as the identity matrix. In this example, the signs of the gradients for the variable metric update have been switched to account for the fact that this is a maximization problem.

Step 1: Given the initialization, iteration 0 has been performed and the following new estimate of the optimum has been found:

$$HP_{\text{Park}} = 85 \text{ nm}$$

$$HA_{\text{Park}} = 143.0478 \text{ nm}$$

$$\omega_{p\text{Park}} = 133.0736 \text{ deg}$$

Step 2: Solve the maximum throw weight problem for the booster to obtain.

$$TW_B^* = 37{,}722.4598 \text{ lb}$$

$$\nu_{\text{Park}} = 341.0832 \text{ deg}$$

$$\partial TW_B^*/\partial HP_{\text{Park}} = -19.3034 \text{ lb/nm}$$

$$\partial TW_B^*/\partial HA_{\text{Park}} = -8.96405 \text{ lb/nm}$$

$$\partial TW_B^*/\partial \omega_{p\text{Park}} = -0.116095 \text{ lb/deg}$$

Step 3: Solve the upper-stage maximum orbital transfer problem, subject to the booster throw weight capability, using the park orbit defined in steps 1 and 2 to obtain

$$TW_U^* = 5117.953 \text{ lb}$$

$$\partial TW_U^*/\partial HP_{\text{Park}} = 0.58809 \text{ lb/nm}$$

$$\partial TW_U^*/\partial HA_{\text{Park}} = 1.86649 \text{ lb/nm}$$

$$\partial TW_U^*/\partial \omega_{p\text{Park}} = 0.55227 \text{ lb/deg}$$

$$\partial TW_U^*/\partial \nu_{\text{Park}} = -0.15336 \text{ lb/deg}$$

$$\partial TW_U^*/\partial TW_B = 0.12699 \text{ lb/nm}$$

Step 4: Calculate the gradients for the system-level problem.

$$\frac{dW_{SC}}{dHP_{\text{Park}}} = \frac{\partial TW_U^*}{\partial HP_{\text{Park}}} + \frac{\partial TW_U^*}{\partial TW_B}\frac{\partial TW_B^*}{\partial HP_{\text{Park}}} = -1.8625 \text{ lb/nm}$$

$$\frac{dW_{SC}}{dHA_{\text{Park}}} = \frac{\partial TW_U^*}{\partial HA_{\text{Park}}} + \frac{\partial TW_U^*}{\partial TW_B}\frac{\partial TW_B^*}{\partial HA_{\text{Park}}} = 0.72814 \text{ lb/nm}$$

$$\frac{dW_{SC}}{d\omega_{p\text{Park}}} = \frac{\partial TW_U^*}{\partial \omega_{p\text{Park}}} + \frac{\partial TW_U^*}{\partial \nu_{\text{Park}}}\frac{\partial \nu_{\text{Park}}}{\partial \omega_{p\text{Park}}} + \frac{\partial TW_U^*}{\partial TW_B}\frac{\partial TW_B^*}{\partial \omega_{p\text{Park}}}$$

$$= 0.69088 \text{ lb/deg}$$

Step 5: Check for convergence of the system-level problem. If the system-level gradient is small, stop.

Step 6: Update the Hessian approximation for the system-level problem (note step 6 is skipped on the zeroth iteration). Calcuate Δx, $\Delta \nabla f$ for variable metric update

$$y = \Delta x = x_{\text{new}} - x_{\text{old}} = \begin{pmatrix}133.07356\\143.04785\end{pmatrix} - \begin{pmatrix}132.0\\139.5\end{pmatrix} = \begin{pmatrix}1.07356\\3.54785\end{pmatrix}$$

$$z = \Delta \nabla f = \nabla f_{\text{new}} - \nabla f_{\text{old}} = \begin{pmatrix}-0.69088\\-0.72814\end{pmatrix} - \begin{pmatrix}-1.0735\\-3.5478\end{pmatrix} = \begin{pmatrix}0.38268\\2.81971\end{pmatrix}$$

Using an initial Hessian approximation equal to the identity matrix, update the approximation by the SR1[15] [Eq. (38)] update to obtain

$$H = H + \frac{(z - Hy)(z - Hy)^T}{y^T(z - Hy)} \tag{38a}$$

$$H = \begin{bmatrix} 0.85644 & -0.15129 \\ -0.15129 & 0.84054 \end{bmatrix} \tag{38b}$$

Step 7: Estimate new park orbit.

$$\begin{pmatrix}\omega_{p\text{Park new}}\\HA_{\text{Park new}}\end{pmatrix} = \begin{pmatrix}\omega_{p\text{Park old}}\\HA_{\text{Park old}}\end{pmatrix} + \beta s$$

with $\beta = 1$, where s is calculated from

$$s = H^{-1}\nabla f = \begin{pmatrix}0.99123\\1.04469\end{pmatrix}$$

This yields a new estimate of the park orbit of

$$\begin{pmatrix}\omega_{p\text{Park new}}\\HA_{\text{Park new}}\end{pmatrix} = \begin{pmatrix}134.0648 \text{ deg}\\144.0925 \text{ nm}\end{pmatrix}$$

Step 8: Estimate new optimum W_{SC}, TW_B, and $WP2$. A linear prediction of the new all-up weight can be found by

$$W^*_{SC\,\text{new linear}} = W^*_{SC\,\text{old}} + \frac{dW^*_{SC}}{dHAPK}\Delta HA_{\text{Park}} + \frac{dW^*_{SC}}{d\omega_p PK}\Delta\omega_{p\text{Park}} = 5119.399 \text{ lb}$$

After several iterations, the Hessian approximation begins to resemble the true Hessian; thus, a quadratic estimate of the new all-up weight can be found by

$$W^*_{SC\,\text{new quadratic}} = W^*_{SC\,\text{new linear}} + 0.5*(\Delta\omega_{p\text{Park}}, \Delta HA_{\text{Park}})H\begin{pmatrix}\Delta\omega_{p\text{Park}}\\HA_{\text{Park}}\end{pmatrix}$$

$$W^*_{SC\,\text{new quadratic}} = 5118.677 \text{ lb}$$

The new booster throw weight to the new park orbit can be estimated by

$$TW_{B\,\text{new}} \approx TW_{B\,\text{old}} + \frac{dTW_B^*}{dHA_{\text{Park}}}\Delta HA_{\text{Park}} + \frac{dTW_B^*}{d\omega_{p\text{Park}}}\Delta\omega_{p\text{Park}} \approx 37{,}712.9800 \text{ lb}$$

Now an estimate of the new weight of $WP2$ (the propellant loading for SRM2 of the upper stage) can be made using

$$WP2_{\text{new}} = WP2_{\text{old}} + (\Delta TW_B) - (\Delta W_{SC}) = 6146.70336 \text{ lb}$$

where

$$\Delta TW_B = TW_{B\,\text{new}} - TW_{B\,\text{old}} = 9.4798 \text{ lb}$$

$$\Delta W_{SC} = W_{SC\,\text{new}} - W_{SC\,\text{old}} = 0.7237 \text{ lb}$$

Step 9: Use the predictions from step 8 as initial guess for the next iteration; go to step 2.

The estimated performance for the second iteration (from step 8) closely matches the numbers obtained in iteration 2 of the decomposition problem (Tables 1–4).

$$W^*_{SC\,\text{iteration 2}} = 5119.10322 \qquad W^*_{SC\,\text{predicted}} = 5118.67672$$

$$TW^*_{B\,\text{iteration 2}} = 37{,}712.868 \qquad TW^*_{B\,\text{predicted}} = 37{,}712.980$$

$$WP2^*_{\text{iteration 2}} = 6146.165 \qquad WP2^*_{\text{predicted}} = 6146.703$$

Thus, the estimates were close to the optimum values and they served as a good starting point for the optimizations in the second iteration. A good agreement between the predicted and observed spacecraft weight, booster throw weight, and weight of propellant loaded was observed on subsequent iterations, especially after the Hessian approximation for the system-level problem began to converge.

A summary of the performance of the all-up decomposition algorithm to a geostationary mission orbit is presented in Tables 1–3. The last row of the tables presents the sensitivity information corresponding to the solution of the all-up optimum (solved by the traditional approach[6]). The maximum payload value found by the decomposition algorithm was slightly larger than that found by the traditional approach but the decomposition algorithm adjusted extra variables (the pitch for the first SRM burn, and the y component of the first RCS burn) for the upper stage which resulted in the small payload discrepancy.

Table 1 presents a summary of the performance quantities obtained for the system-level problem. The system gradients approached zero and the value of the objective function converged close to the same value as obtained by the traditional formulation. The final row in Table 1 titled "All-up" is the solution (reported in Hallman[6]) to the problem when solved by the all-up formulation [Eqs. (28-33)]. Hallman[6] reports $dW^*_{SC}/dHP_{\text{Park}} = -3.670$ lb/nm, the predicted sensitivity (by the decomposition algorithm) of maximum payload to the perigee of the park orbit matched to several places. The solution obtained by the decomposition algorithm is slightly better

Table 1 Summary of the performance for the system-level problem to geostationary orbit

Iteration	At park orbit insertion					
	HA_{Park}, nm	$\omega_{p\text{Park}}$, deg	W^*_{SC}, lb	$\dfrac{dW^*_{SC}}{dHA_{\text{Park}}}$, lb/nm	$\dfrac{dW^*_{SC}}{d\omega_{p\text{Park}}}$, lb/deg	$\dfrac{dW^*_{SC}}{dHP_{\text{Park}}K}$, lb/nm
0	139.5	132.0	5110.438	3.54791	1.07356	2.36447
1	143.0478	133.0735	5117.954	0.72814	0.69088	−1.86325
2	144.0925	134.0648	5119.103	0.27629	0.58600	−2.61675
3	145.4857	135.5867	5119.913	−0.20981	0.43219	−3.48148
4	147.4469	140.9345	5119.868	−0.91606	−0.05804	−5.00685
5	146.1374	138.5468	5120.629	−0.51894	0.20433	−4.18400
6	145.5767	138.6650	5120.912	−0.39801	0.22800	−3.99276
7	142.3549	144.5584	5122.239	−0.05516	0.03944	−3.56406
8	142.4308	144.8345	5122.248	0.00883	0.00077	−3.66555
9	142.3796	144.9891	5122.249	0.00129	0.00188	−3.67027
All-up[a]	142.3377	145.0790	5122.181	0.00370	0.00058	−3.66938

[a]Note: This is the solution reported in Hallman[6] where Eqs. (28–33) were used to define the problem.

Table 2 Summary of the performance for the upper-stage problem to geostationary orbit

Iteration	HA_{Park}, nm	$\omega_{p\text{Park}}$, deg	ν, deg	TW^*_U, lb	$WP2$, lb	$\dfrac{dTW^*_U}{dTW_B}$, lb/lb	$\dfrac{dTW^*_U}{dHA_{\text{Park}}}$, lb/nm	$\dfrac{dTW^*_U}{d\omega_{p\text{Park}}}$, lb/deg	$\dfrac{dTW^*_U}{d\nu}$, lb/deg
0	139.5	132.0	342.157	5110.44	6196.18	−0.08512	2.78675	0.83482	−0.24908
1	143.048	133.074	341.083	5117.95	6156.91	0.12699	1.86648	0.55227	−0.15336
2	144.093	134.065	340.090	5119.10	6146.17	0.16122	1.72572	0.49909	−0.13797
3	145.486	135.587	338.920	5119.91	6132.06	0.19765	1.57763	0.43961	−0.12156
4	147.447	140.935	333.206	5119.87	6107.26	0.24719	1.39360	0.32880	−0.09927
5	146.137	138.547	335.599	5120.63	6122.52	0.22002	1.50148	0.38160	−0.11147
6	145.577	138.665	335.481	5120.91	6127.24	0.21129	1.54298	0.38888	−0.11545
7	142.355	144.584	329.572	5122.24	6145.02	0.17720	1.75364	0.34316	−0.13053
8	142.431	144.834	329.295	5122.25	6143.59	0.18081	1.73861	0.33648	−0.12887
9	142.380	144.989	329.140	5122.25	6143.70	0.18065	1.74066	0.33447	−0.12892
All-up	142.338	145.079	329.125	5122.18	6143.94	0.18222	1.74183	0.33417	−0.12906

Table 3 Summary of the performance for the booster problem to geostationary orbit

Iteration	HA_{Park}, nm	$\omega_{p\text{Park}}$, deg	TW^*_B, lb	$\dfrac{dTW^*_B}{dHA_{\text{Park}}}$, lb/nm	$\dfrac{dTW^*_B}{d\omega_{p\text{Park}}}$, lb/deg
0	139.5000	132.0	37754.21	−8.94169	−0.12153
1	143.0478	133.0735	37722.56	−8.96405	−0.11609
2	144.0935	134.0648	37712.87	−8.99062	−0.31677
3	145.4857	135.5867	37699.58	−9.04354	−0.65252
4	147.4469	140.9345	37674.73	−9.34364	−1.96653
5	146.1374	138.5468	37690.75	−9.18287	−1.31233
6	145.5767	138.6650	37695.75	−9.18633	−1.30780
7	142.3549	144.5844	37714.86	−9.58520	−2.45067
8	142.4308	144.8345	37713.45	−9.61154	−2.52491
9	142.3796	144.9891	37713.55	−9.62525	−2.55803
All-up	142.3377	145.0790	37713.72	−9.63314	−2.52943

Table 4 Summary of predicted performance for the system-level problem to geostationary orbit

Iteration	W^*_{SC}, lb	W^*_{SC} (predicted), lb	β (cubic)	W^*_{SC} (β cubic prediction), lb
0	5110.483	5112.748	1.13119	5118.180
1	5117.954	5118.676	3.04448	5119.922
2	5119.103	5119.742	1.45021	5119.994
3	5119.913	5120.862	0.28003	5120.152
4	5119.868	5120.537	1.17126	5120.646
5	5120.629	5120.786	4.15802	5121.325
6	5120.912	5122.225	1.02197	5122.240
7	5122.239	5122.263	0.49804	5122.248
8	5122.248	5122.249	1.05713	5122.249
9	5122.249	5122.249	—	—

than the solution obtained by Hallman but the variance is within numerical tolerance. The spacecraft weight decreased between iterations 3 and 4; this is due to the simplicity of the optimization algorithm used to solve the system-level problem.

Table 2, a summary of the upper-stage performance, shows that the true anomaly at park orbit insertion decreased. The starting point was far from the optimum; this caused dTW^*_U/dTW_B to be less than zero, which is counterintuitive. The fixed size of SRM1 for the upper stage means that the upper stage was not optimally sized for the park orbit on the zeroth iteration.

Table 3, a summary for the booster problem, shows the throw weight oscillated in the first few iterations but then converged to an optimal value, $\partial TW^*_B/\partial HA_{\text{Park}}$ was fairly constant, and $\partial TW^*_B/\omega_{p\text{Park}}$ decreased as the argument of perigee increased.

By optimizing the all-up problem, an additional 11.811 lb of payload can be placed into a geostationary mission orbit as compared to using the standard park orbit (85 × 139.5 nm with $\omega_p = 132$ deg). It is also observed that the region of the optimum is rather flat. Thus, if there is any variance in the weight of propellant for SRM2 ($WP2$) due to manufacturing, then there is a small penalty in performance if the all-up trajectory profile is reoptimized.

The final Hessian approximation for the system problem was found to be

$$H = \begin{bmatrix} 0.08749 & 0.11689 \\ 0.11689 & 0.37445 \end{bmatrix}$$

This represents an approximation to the following sensitivity derivatives

$$H = \begin{bmatrix} \dfrac{\partial^2 W_{SC}}{\partial \omega_{p\text{Park}}^2} & \dfrac{\partial^2 W_{SC}}{\partial \omega_{p\text{Park}} \partial HA_{\text{Park}}} \\ \dfrac{\partial^2 W_{SC}}{\partial HA_{\text{Park}} \partial \omega_{p\text{Park}}} & \dfrac{\partial^2 W_{SC}}{\partial HA_{\text{Park}}^2} \end{bmatrix}$$

The Hessian indicates that the system-level problem may be poorly scaled. Some scaling of the system-level problem [Eqs. (34–37)] may yield better performance of the decomposition algorithm. This information can be used to predict the effect of flying to a nonoptimal park orbit by using the following approximation:

$$W_{SC}^*(\Delta HP_{\text{Park}}, \Delta HA_{\text{Park}}, \Delta\omega_{p\text{Park}}) \approx \frac{\partial W_{SC}^*}{\partial HP_{\text{Park}}} \Delta HP_{\text{Park}}$$

$$+ 0.5(\Delta\omega_{p\text{Park}}, \Delta HA_{\text{Park}}) \begin{bmatrix} 0.08749 & 0.11689 \\ 0.11689 & 0.37445 \end{bmatrix} \begin{pmatrix} \Delta\omega_{p\text{Park}} \\ \Delta HA_{\text{Park}} \end{pmatrix}$$

where ΔHP_{Park}, ΔHA_{Park}, $\Delta\omega_{p\text{Park}}$ represent the perturbations in the park orbit. The $\partial W_{SC}^*/\partial HA_{\text{Park}}$ and $\partial W_{SC}^*/\partial \omega_{p\text{Park}}$ terms are not in the preceding equation because they are equal to zero at the solution. If additional constraints on ground tracking or park orbit free molecular heating were imposed on the problem, then the above equation could be used to estimate the payload penalty associated with flying to a nonoptimal park orbit. Finally, Table 4 is presented to illustrate how the predicted performance agreed with the observed performance. The first column in Table 4 represents the observed maximum spacecraft weight on each iteration. The second column is the predicted spacecraft weight on the next iteration using a quadratic prediction that includes the Hessian approximation. The column headed with β(cubic) is a cubic prediction of the optimal step length in the line search that was created using directional derivatives. The final column of Table 4 is the predicted maximum objective function value in the search direction using a cubic approximation. It can be seen that several of the predicted optimal step lengths were close to one, therefore the crude line search that was used should suffice, but there is still some room for improvement. Also, the predicted spacecraft weight using a quadratic model began to yield good approximations after several iterations were performed and the Hessian approximation began to resemble the true Hessian.

Case 2: All-Up Performance to a 1-Deg Geosynchronous Mission Orbit

An all-up trajectory optimization to a 1-deg geosynchronous mission orbit was solved, and the results are summarized in Table 5 and Ref. 17. The final row of Table 5 provides the sensitivity derivatives obtained if the solution to the all-up problem (found using the traditional formulation [Eqs. (28–33)], see Hallman[6]) is used as the starting point for the decomposition algorithm. The gradient of the system-level problem is near zero on the fifth iteration. The all-up maximum payload predicted by the decomposition algorithm, and that found by the traditional formulation, agree to seven places. An additional 4.23 lb of payload capability were obtained (over the starting point) by optimizing the all-up trajectory. The starting point for this case was close to the solution: thus less work was required to solve the problem.

The final Hessian approximation for the system-level problem is

$$H = \begin{bmatrix} 0.06405 & 0.05418 \\ 0.05418 & 0.20816 \end{bmatrix}$$

This approximation is similar to the approximation found in the geostationary case. There was less change in the Hessian approximation during the last few iterations than that observed for the geostationary case. Thus, the Hessian approximation for the 1-deg geosynchronous case may be more accurate than the Hessian approximation found in the geostationary case; however, the system-level problem seems to be rather nonlinear and scaling the problem may reduce the number of system-level iterations required for solution.

Future Work

Several areas where this technique is still in need of development are 1) formulations for other mission scenarios, 2) the ability to deal with park orbit heating constraints, 3) efficient algorithms for parameter sensitivity analysis, 4) the effect of nonconvergence of the booster or upper-stage problems on the accuracy of the sensitivity derivatives, 5) a hot start for the booster problem, 6) convergence criteria and scaling for the system-level problem, 7) methods to calculate $dW_{SC}^*/d(\,)$ where ( ) are design parameters, and 8) computational testing. Some of the previous issues are now addressed.

This paper has described the solution of the maximum payload problem with upper-stage sizing and a fixed park orbit perigee altitude. Other formulations are required to solve the maximum payload problem with fixed upper-stage weight, or maximum reserve fuel trajectories for fixed payload weights. Some other formulations have been investigated and

Table 5 Summary of performance for the system-level problem to 1-deg geostationary orbit

Iteration	At park orbit insertion HA_{Park}, nm	At park orbit insertion $\omega_{p\text{Park}}$, deg	W_{SC}^*, lb	$\dfrac{dW_{SC}^*}{dHA_{\text{Park}}}$, lb/nm	$\dfrac{dW_{SC}^*}{d\omega_{p\text{Park}}}$, lb/deg.	$\dfrac{dW_{SC}^*}{dHP_{\text{Park}}}$, lb/nm
0	143.0478	132.8244	5171.632	0.63005	0.68277	−2.02094
1	143.6707	133.5072	5172.374	0.33919	0.61612	−2.49818
2	145.2320	136.3433	5173.808	−0.29088	0.37634	−3.67604
3	145.3230	138.6614	5174.467	−0.44489	0.21635	−4.08187
4	142.2233	144.5270	5175.856	−0.01500	0.01915	−3.69316
5	142.0585	144.9733	5175.862	−0.00488	−0.00050	−3.69570
All-up	142.0478	145.0042	5175.862	0.00060	0.00058	−3.68796

have yielded encouraging results, but more testing is required to refine these formulations.

Often when all-up trajectory optimization problems are solved, there are some constraints on the free molecular heating in the park orbit due to heating limits on the spacecraft. These constraints can be added to the traditional formulation with ease. In the decomposition problem, an additional constraint is required for the system-level problem because the free molecular heating is a function of the park orbit apogee, perigee, argument of perigee, and the weight of the vehicle.

The calculation of parameter sensitivity derivatives is the key to successful convergence of the decomposition algorithm, but they can be computationally expensive to find and are sensitive to the convergence criteria for the booster and upper-stage problems. Tight convergence criteria often require extra iterations for the optimization algorithm to converge. For the booster problems, there is a certain level below which the convergence criteria cannot be satisfied because of noise in the problem. On the first few iterations of the system problem, it is not absolutely necessary to have exact values for the parameter sensitivity derivatives, but as the algorithm converges, more accurate sensitivity derivatives are required. Thus, the convergence criteria for the optimzation algorithm could be set loose for the first few iterations and then tightened up when the system-level problem begins to converge.

Beltracchi[8] found that when solving perturbed problems using the RQP algorithm, certain information which was built up during the solution of the baseline problem (such as active set and Hessian approximations) can be used to increase the speed of the RQP method. This is called using a hot start for solving the perturbed problem. Using parameter sensitivity derivatives to generate an estimate of the optimum of the perturbed problem to be used as an initial guess for the RQP method reduces the solution time for the perturbed problem. A hot-start procedure could be created for the GPG algorithm to further improve the performance of the algorithm.

Using a hot start to solve the booster and upper-stage problems means that in successive iterations the solution of the subproblems will require less CPU time; thus convergence of the decomposition algorithm should be accelerated. The evaluation of the gradients of the objective function and constraints with respect to the design variables for the booster accounts for a large part of the computational expense of solving the booster problem. On successive system iterations the gradients for the booster problem are known at the base point and only the right side of the constraints change, thus the gradients from the previous booster problem may be reused (if the starting point is taken as the optimum point from the previous iteration) resulting in further savings.

From the two examples studied in this paper, it became apparent that some scaling for the system-level problem is required for faster convergence of the algorithm. The proper scaling for the system-level problem is unknown. After each all-up optimization problem is solved, the solution could be stored in a data base and the starting point for similar cases could be taken as an interpolation of the best available data. A convergence criteria could be set by limiting the predicted increase in system-level performance and comparing the predicted increase in the spacecraft weight to that tolerance.

Finally, when the all-up optimization problem is solved, parameter sensitivity derivatives can be calculated for any quantity that is specified in the parameter list (i.e., dry weights of the stages, specific impulse of the engines, reserve fuel). There should be some corresponding formulas for the decomposition algorithm that allow the same type of information to be estimated.

Conclusions

This paper has described a decomposition algorithm that is capable of solving the all-up trajectory optimization problem. The algorithm is based on solving separate booster and upper-stage trajectory optimization problems. The algorithm can be applied to obtain an approximation of the all-up performance or for generating near optimal starting points for solving the traditional all-up problem, by terminating the decomposition algorithm after only a few iterations. The algorithm can also be used to obtain a prediction of how the payload capability changes for nonstandard park orbits.

Acknowledgments

The author would like to thank S. M. Hammes, R. A. Luke, and P. Oliphant of the Aerospace Corporation for initial technical reviews, and also the AIAA technical reviewers for their comments.

References

[1]GTS, "The Generalized Trajectory Simulation System," Vol. 1–5, Rept. TR-0076(6666)-1, Aerospace Corp., El Segundo, CA, April 1976.

[2]Brauer, G. L., Cornick, D. E., Olson, D. W., Petersen, F. M., and Stevenson, R., "Program to Optimize Simulated Trajectories (POST)," Vols. 1–3, Rept. MCR-87-583, NAS1-18147," Martin Marietta Corp., Denver CO, Sept. 1987.

[3]Sobieszczanski-Sobieski, J., James, B. B., and Riley, M. F., "Structural Sizing by Generalized, Multilevel Optimization," *AIAA Journal,* Vol. 25, No. 1, 1987, pp. 139–145.

[4]Fiacco, A. V., *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming,* Academic Press, New York, 1983.

[5]Hallman, W. P., "A Software Package for Post Optimality Analysis of the Nonlinear Programming Problem," Aerospace Corp., El Segundo, CA, Internal Documentation, April 1986.

[6]Hallman, W. P., "Sensitivity Analysis for Trajectory Optimization Problems," AIAA Paper 90-0471, presented at the Aerospace Sciences Meeting and Exhibit, Reno, NV, Jan. 1990.

[7]Beltracchi, T. J., and Gabriele, G. A., "An Investigation of New Methods for Estimating Parameter Sensitivities," NASA CR 4245, Aug. 1989.

[8]Beltracchi, T. J., "An Investigation of Parameter Sensitivity Analysis by the Recursive Quadratic Programming Method," Ph.D. Thesis, Rensselaer Polytechnic Inst., Troy, NY, Aug. 1988.

[9]Betts, J. T., and Hallman, W. P., "NLP2 Optimization Algorithm Documentation," Aerospace Corp., TOR-0089 (4464-06)-1, El Segundo, CA, Aug. 1989.

[10]Hallman, W. P., "Optimal Scaling Techniques for the Nonlinear Programming Problem," Aerospace Corp., El Segundo, CA, Internal Documentation, June 1989.

[11]Luke, R. A., "Computational Efficiency Considerations for High Fidelity Launch Vehicle Trajectory Optimization," *Proceedings of the AIAA Guidance, Navigation, and Control Conference,* AIAA, Washington, DC, 1989.

[12]Huffman, W., "An Analytic J_2 Propagation Method for Low Earth Orbits," Aerospace Corp., El Segundo, CA, Internal Documentation, Nov. 1981.

[13]Luke, R. A., personal communication, Sept. 1989.

[14]Petersen, F. M., Cornick, D. E., and Brauer, G. L., "A Two Level Trajectory Decomposition Algorithm Featuring Optimal Intermediate Target Selection," *Journal of Spacecraft and Rockets,* Vol. 14, No. 11, 1977, pp. 676–682.

[15]Beltracchi, T. J., and Gabriele, G. A., "A Hybrid Variable Metric Update for the Recursive Quadratric Programming Method," *Advances in Design Automation: Vol. 2, Design Optimization,* ASME DE 19-2, 1989; also *Journal of Mechanical Design,* Vol. 113, No. 3, 1991, pp. 280–285.

[16]Powell, M. J. D., "How Bad Are the BFGS and DFP Methods When the Objective Function is Quadratic," *Mathematical Programming,* Vol. 34, 1986, pp. 34–47.

[17]Beltracchi, T. J., "A Decomposition Approach to Solving the All-Up Trajectory Optimization Problem," AIAA Paper 90-0469, Reno, NV, Jan. 1990.