Covariance Matrix Adaptation Evolution Strategy for Multidisciplinary Optimization of Expendable Launcher Family

Guillaume Collange[1] and Stéphane Reynaud[2]
EADS Astrium Space Transportation, Les Mureaux, Ile-de-France, 78133, France

and

Nikolaus Hansen[3]
INRIA Saclay - Ile-de-France LRI, Orsay, Ile-de-France, 91405, France

After 30 years of success of Ariane launches, Astrium Space Transportation as prime contractor is preparing the future of launch vehicles with research and development activities. This paper describes the results of the collaboration between INRIA and Astrium to solve the typical multidisciplinary problem of expendable launch vehicle design thanks to the Covariance Matrix Adaptation Evolution Strategy (CMA-ES). The different disciplines integrated in the Multidisciplinary platform are propulsion system, aerodynamics, mass budget, trajectory integration, control. CMA-ES was tested on a two-liquid-staged launcher with solid boosters. The algorithm produced conclusive results on an optimization problem that proved to be very ill-conditioned. The comparison with Non-Dominated Sorting Genetic Algorithm NSGA-II gave equivalent results on a bi-level optimization, the trajectory sub-problem being solved separately by a reduced gradient method. The good performance of CMA-ES on a single launcher case allowed us to extend the tests on a launcher family. A launcher family is composed of several launcher configurations sharing common characteristics with different payload targets and optimized together. In these last cases, CMA-ES surpasses NSGA-II in terms of performance and was able to handle multiple error cases during the search of optimum.

Nomenclature

A5	=	Ariane 5
CMA-ES	=	Covariance Matrix Adaptation – Evolution Strategy
GTO	=	Geostationnary Transfer Orbit
GLOW	=	Global Lift-Off Mass
INRIA	=	Institut National de Recherche en Informatique et Automatique
IRFNA	=	Inhibited Red Fuming Nitric Acid
MCI	=	Mass, Center of gravity, Inertia
MDO	=	Multi-Disciplinary Optimization/Optimisation Multi-Disciplinaire (OMD)
MMH	=	Mono-Methyl Hydrazine
NSGA	=	Non-Sorting Genetic algorithm
RNTL	=	Réseau National Telecom et Réseau
UDMH	=	Unsymmetrical Di-Methyl Hydrazine
TVC	=	Thrust Vector Control

[1] Aerospace Engineer, Mission Analysis & Navigation Systems Department, guillaume.collange@astrium.eads.net

[2] Head of Mission Analysis section, Mission Analysis & Navigation Systems Department, stephane.reynaud@astrium.eads.net

[3] Senior Researcher, Machine Learning and Optimization group (TAO), nikolaus.hansen@lri.fr

AoA	=	Angle-of-Attack
m	=	mass
Q	=	mass flow rate
T	=	Thrust
Is	=	Specific Impulse
$\mathbf{X}$	=	launcher position vector
$\mathbf{V}$	=	launcher velocity vector
V	=	$\mathbf{V}$ modulus
$\mathbf{V}_r$	=	launcher relative velocity vector (absolute velocity minus Earth training velocity)
V_r	=	$\mathbf{V}_r$ modulus
γ	=	launcher acceleration vector
$\mathbf{g}$	=	gravity acceleration vector
g_0	=	9.80665 m/s²
θ	=	launcher pitch orientation in a given galilean frame
ψ	=	launcher yaw orientation in a given galilean frame
ρ	=	density
C_D	=	drag coefficient
S_{ref}	=	Reference area
Pdyn	=	Dynamic Pressure ($0.5\rho V_r^2$)
Φ	=	Heat flux ($0.5\rho V_r^3$)
c	=	chord
f, g	=	generic functions
h	=	height

I. Introduction

AFTER 30 years of success of Ariane launches, Astrium Space Transportation as prime contractor is preparing the future of launch vehicles with research and development activities. This paper describes the results of the collaboration between INRIA and Astrium to solve the typical multidisciplinary problem of expendable launch vehicle design thanks to the Covariance Matrix Adaptation Evolution Strategy (CMA-ES).

During the preliminary phases of the development of a new launch vehicle, several disciplines have to be considered in order to obtain a first coherent design. Each subject is usually treated separately. As these disciplines have strong interaction between each other, the development leads to numerous loops between the dedicated teams. We are facing a typical multidisciplinary problem. Without treating it globally, an optimal solution could be difficult to reach. The multidisciplinary optimization (MDO) concept takes into account the constraints of different disciplines with the aim of producing the optimized design with respect to high level criteria.

Working on MDO principle, Astrium has developed a software platform gathering the relevant disciplines for the preliminary design of launch vehicles. The different modules or meta-models, corresponding to disciplines identified as critical, have been simplified from heavier tools. The different modules are integrated into a single environment. The driving parameter to develop these modules is the computation time. Based on this platform, the parameters, constraints and criteria of the multidisciplinary optimization can be defined. Different solutions[1-4] have already been explored since 2004. With the numerous parameters and strict constraints of the problem, very difficult optimization problems arise. Some other approaches[5-7] using various evolutionary techniques have been explored. This paper focuses on a new algorithm CMA-ES to solve the launcher design problem and extend it to solve launcher family design problem.

The "RNTL/OMD" project is a collaboration between universities, research laboratories and industrials on "Multidisciplinary Optimization". This project was selected by the French network for research and innovation on software technologies (Réseau National recherche et d'innovation en Technologies Logicielles "RNTL") as an exploratory project. In this frame, the French Research Agency (Agence National de la Recherche "ANR") funded part of the activities of the different partners. In the frame of the "RNTL/OMD" project, Astrium has proposed different test cases on preliminary design of launch vehicles. They were generated from our MDO platform with different level of complexity.

II. Multidisciplinary Platform

Over the years, Astrium Space Transportation has developed specific tools integrated in a multidisciplinary software platform. They were based on standard software used in advanced project studies. These tools were adapted especially in terms of interfaces to fit the global integration and optimization purpose. The propulsion/structure module provides the main engine characteristics as well as mass and inertia of solid or liquid propellant stages. The aerodynamics module provides drag and lift coefficients. The mission/trajectory tool integrates the equation of motion depending on launcher characteristics and command law. The cost module is representative of the manufacturing cost of the launcher.

A. Propulsion and stage mass budget module

The propulsion and stage mass budget module is adapted from existing tools in the propulsion department. Each stage is treated separately depending on the propulsion type. The PROPSOL tool is dedicated to solid propellant stages and the PROPLIQ tool is dedicated to liquid propellant stages. Even if these tools are used in advanced projects, they have lots of inputs in order to answer all possible concepts studied. For the most detailed studies, the number of inputs can be close to a hundred. In the frame of the multi-optimization platform, the number of inputs has to be reduced to allow optimization. The number of optimized parameters should be limited to relevant ones i.e. the retained parameters should have a significant impact on the design.

For solid propulsion, we kept all the parameters needed for the vehicle mission analysis, i.e. restricted to the following inputs: propellant mass, average mass-flow rate, diameter, low or upper stage and case material choice. All the other values are conventional depending on the stage configuration: in terms of nozzle design, average pressure value (linked with the kind of grain), propellant kind, maximum pressure levels (linked with case materials), margins, steering angle… The outputs are stage mass budget and performance parameters: overall inert mass (case, thermal protection, nozzle assembly, TVC devices and other stage components) as well as specific impulse and nozzle exit area. The stage layout is also available, mainly the length, which is used to determine slenderness and the inertia for the controllability module.

For liquid propulsion, in addition to parameters already considered for solid propellant stages - propellant mass, average mass-flow rate, stage diameter, kind of stage – specific parameters have been added: oxidizer/fuel kind and tank architecture. Available propellants cover a wide range of technologies: N2O4, IRFNA and LOX coupled with MMH, UDMH, kerosene (here RP1), LH2 and CH4. The tanks may be tandem tanks (with common or separated bulkhead) or in parallel. All other parameters are chosen from state of the art (may be optimized separately with the stand alone code version): mixture ratio, pressurization devices, nozzle area ratio, materials and engine cycle. The outputs are the same as PROPSOL, i.e. stage mass budget and performance parameters: overall inert mass (tanks, engines, thermal protection, pressurization systems, TVC devices and other stage components) as well as specific impulse and nozzle exit area.

B. Aerodynamics Module

The aerodynamic coefficients necessary for the trajectory assessment are computed through specific methods that have been developed to allow rapid evaluation of the vehicle performance in the frame of preliminary projects. These methods allow assessing the aerodynamic characteristics of launchers or missiles, around zero angle-of-attack, for Mach numbers during atmospheric phase. The geometries that can be taken into account (an example is given figure 6) are axisymmetric launchers with or without boosters (up to four), with 1 up to 4 stages, composed of conical or ogival nosecap, cylinder(s), skirt(s) or boattail(s),empennages, base with nozzles.

The main outputs of the tool are the drag and lift coefficients of the global launcher as well as the position of the application point of the aerodynamic force.

C. Trajectory modules

Two different trajectory modules have been implemented in the Multidisciplinary platform depending on the approach[4] to solve the launcher design problem. The main difference is the integration of the equation of motion. The first software called TRAJ1[8] is based on a numerical integrator (typically 7th/8th order Runge-Kutta) and the second one called TRAJ2 is based on semi-analytical integration of the equation, which provides faster results but less precise. Both tools solve the following system representing the motion of the launcher center of gravity position and velocity from lift-off to payload insertion:

$$\begin{cases} \dot{\mathbf{X}} = \mathbf{V} \\ \dot{\mathbf{V}} = \boldsymbol{\gamma} \\ \dot{m} = -Q \\ \boldsymbol{\gamma} = \mathbf{g} + (\mathbf{F}_{aero} + \mathbf{F}_{prop})/m \\ \mathbf{F}_{aero} = -0.5\rho V_r{}^2 C_D S_{ref} (\mathbf{V}_r / V_r) \\ \mathbf{F}_{prop} = T(\cos\theta\cos\psi\mathbf{i} + \cos\theta\sin\psi\mathbf{j} + \sin\theta\mathbf{k}) \end{cases} \tag{1}$$

The first order system of equation represented here is a typical mechanical problem based on Newton principle. The acceleration of the center of mass of the launcher is due to three main forces, gravitational **g**, aerodynamic $\mathbf{F}_{aero}$ and propulsive $\mathbf{F}_{prop}$. The propulsive force is expressed with three parameters, the modulus T is the thrust of the engine and two angles θ et ψ representing the command law of the launcher in reference galilean frame (**i,j,k**) linked to the launched pad location at lift-off. The nine-dimension state (**X**,**V**,m,θ,ψ) has to be integrated with respect to time from lift-off to payload injection in order to obtain the launcher trajectory. Theoretically, the command law should be assessed as time functions θ(t) and ψ(t) by the optimizer. In practice, in order to keep commonality with on-board guidance algorithms, these laws are cut into segments whose beginning and end are defined according to trajectory events (like engine ignition or shut-off, path constraint, mass jettison). For each segment, θ and ψ are supposed to be time-linear functions, therefore fully defined by values at segment bounds.

In a typical trajectory optimization problem[1], the main outputs of the trajectory considered are the constraints of the optimization. These constraints can be divided into two groups: final orbit constraints and intermediate constraints.

The final orbit constraints are usually specified by the customer depending on the payload mission (geostationary orbit for telecommunication, low earth orbit for observation…). The final position and velocity constraints are specified in terms of orbital elements (apogee altitude, perigee altitude, inclination, perigee argument…).

The intermediate constraints are either linked to environment loads that the launcher or the payload can stand (maximum dynamic pressure, heat flux or acceleration), or to operational requirements (safeguard of the launch pad, stage fall-out zone, visibility of the launcher…).

The trajectory module cannot generate correct trajectories for each set of input. We should deal with a lot of error cases. Among them, the launcher does not have enough thrust to lift-off, the launcher does not have enough energy to orbit the payload or crash before reaching orbit. An error case management system is then integrated into the module.

D. Controllability

An analytical method to assess the controllability of the launcher has been developed for the MDO platform. During its atmospheric flight, a generic launcher is usually unstable due to a center of mass lower than the application point of the aerodynamic force. When the launcher is in the atmosphere, the nominal command law considered a zero angle-of-attack attitude. But in reality, the dispersions and the wind perturb the attitude and the thrust vector control system compensates the instability of the launcher due to Pdyn.AoA force. The controllability module computes the static and dynamic nozzle actuation angle needs. The static nozzle actuation angle is assessed from the dispersions on the direction and point of application of the thrust with respect to the center of gravity. The dynamic nozzle actuation angle is assessed from the reaction of the launcher to a typical gradient and gust of wind. The outputs of the module are the maximum nozzle actuation angle and angular velocity. In the MDO platform, we can apply an inequality constraint on these values to limit them and produce a launcher feasible in terms of control system need.

E. Recurrent cost evaluation

Different cost models can be implemented in the platform depending on the information available. For the purpose of this study only the launcher recurrent cost was taken into account. A unit stage cost Ci is expressed by $Ci = \alpha M^{\beta}$, where M is either the engine or the stage dry mass (according to the type of technology considered) and α and β are coefficients adapted to the considered stage (again depending on the type of technology itself represented by 3 integer parameters in our 28 parameter problem, see above). The total cost C is then simply the sum of Ci over the number of stages.

F. Multidisciplinary data flow

Figure 1 depicts the implementation of the different modules inside the MDO platform as well as the data flow between them. The main input and output of the platform are also represented. The inputs are the potential degrees of freedom or parameters of the multidisciplinary optimization problem and the output can be defined as constraints or criterion of the problem. Different optimization algorithms can be plugged in the platform.

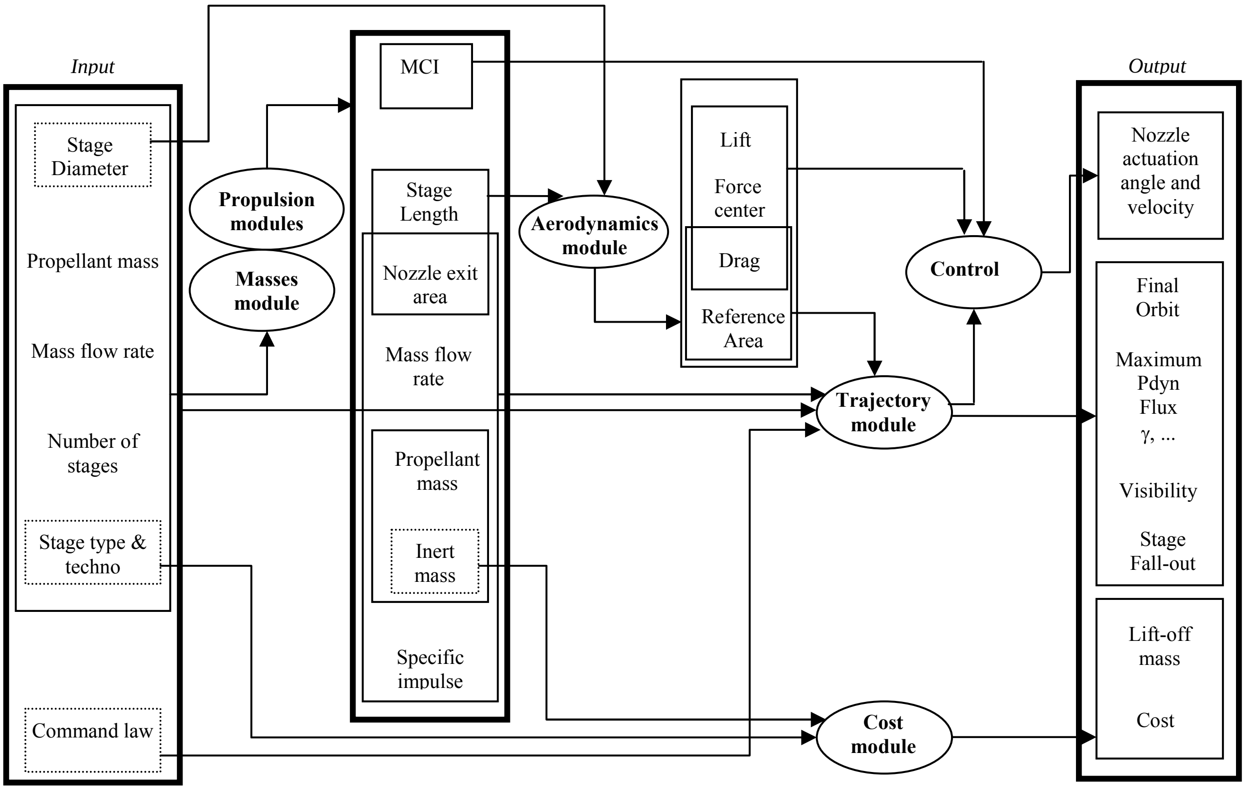

Figure 1 : MDO platform data flow

III. Covariance Matrix Adaptation Evolution Strategy (CMA-ES)

Evolutionary algorithms are optimization (or search) methods based on principles found in biological evolution, in particular recombination, mutation, and selection. They are commonly used for non-linear black-box optimization, where no specific knowledge about the problem is available, but the cost function can be repeatedly evaluated. Evolution Strategies[9] are Evolutionary Algorithms specialized for searching in continuous, multidimensional spaces. A recent variant is the so-called Covariance Matrix Adaptation Evolution Strategy (CMA-ES)[10,11]. The "default" CMA-ES is a non-elitist algorithm that first samples a number of new candidate solutions from a multivariate normal distribution and then updates the sampling distribution only using the better (new) solutions. The update consists of two major mechanisms: step-size control and covariance matrix adaptation. For step-size control, the length of the path of the most recent iteration steps is analyzed (path length control or cumulative step-size adaptation). For covariance matrix adaptation the likelihood of successful steps is increased. The time scale of both updates is essentially different. The step-size can change fast, in order to allow for fast convergence to a good solution. The covariance matrix only changes on a slower times scale in order to foster its stability.

The CMA-ES has shown superior performance in several benchmarking exercises[12,13] and was already applied successfully to a number of real-world problems. Compared to other evolutionary and non-evolutionary search algorithms, the CMA-ES performs in a particularly superior way on difficult, non-separable problems, where the number function evaluations, needed to find a satisfactory solution, exceeds 100 times the search space dimension; for example, ill-conditioned, rugged or noisy functions with search space dimension not much smaller than ten.

In this paper, we use a novel constraints handling technique, epsilon-normalized success-controlling penalty constraint handling, with CMA-ES in order to solve the launcher test cases presented below. The constraints handling is applicable to non-linear equality and inequality constraints and can be combined with most evolutionary algorithms. The main idea is to control the minimum number of feasible candidate solutions generated by the search procedure. The actuating variables are weight factors used in a penalty term that is added to the cost function. In short, if the number of feasible solutions becomes too small, the penalization weights of the respective constraints are increased. The constraint handling requires first to define, for each constraint, an acceptable error threshold, epsilon.

- For inequality constraints, epsilon determines when the penalization term starts to be applied, even when the constraint value still indicates a feasible solution. In this way the continuity of the constraint value is exploited. For a 0/1 indicator constraint value, the epsilon value becomes irrelevant.
- For equality constraints, epsilon determines when a solution is regarded as infeasible. Because the sampling procedure is stochastic and admits a density, equality constraints cannot be satisfied exactly. Equality constraints are consequently always penalized.

For each constraint, the penalty weight is increased, if the ratio of feasible solutions, in the given iteration and averaged over some iteration steps, is smaller than a target value and the number of feasible solutions has not increased over the last four iterations. The resulting weights allow to search in the infeasible domain, while always a few feasible solutions are maintained. This policy is described in more detail elsewhere[14]. It has appeared to be useful on artificial test functions and was applied to the Astrium test cases.

IV. Launcher Test Cases

A. Test cases presentation

The goal of launcher test cases is to optimize a two liquid stage launcher with two solid boosters injecting 8 tons of payload in GTO. The architecture of the launcher is fixed with the following parameters:

- The boosters referenced (1) in Figure 2 are considered with solid propellant and metallic cases.
- The main central stage (2) is propelled with a cryogenic engine using liquid hydrogen and liquid oxygen. The architecture of the stage is built of tandem tanks with common bulkhead.
- The upper stage (3) works with storable N_2O_4/MMH propellants and tandem tanks with separate bulkhead.

The injected payload for the launcher is 8 tons and the final orbit targeted is a geostationary transfer orbit (GTO) defined by the following orbital parameters:

Parameters targeted	GTO
Apogee altitude	35 786 km
Perigee altitude	≥ 200 km
Inclination	7°
Perigee argument	178°

Table 1 : GTO final orbit constraints

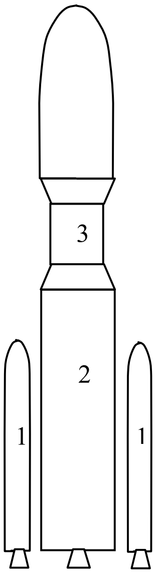

Figure 2 : launcher test case

On the two test cases presented in the paper, CMA-ES algorithms is tested on different complexity of problem in terms of number of parameters and on different modes of the MDO platform. The first case is a simplified one where the trajectory module is replaced by an equivalent propulsive velocity increment corresponding to GTO orbit. The second test case is more complex with additional parameters for the command law and the main trajectory constraints.

B. First Test case

The first test case is performed without the trajectory module and the global performance of the launcher is represented by an equivalent propulsive velocity increment (ΔV). The 9 optimized parameters are the mass flow rates, the propellant masses and the diameters of the 3 stages. The optimization criterion is the recurrent cost of the launcher under 16 inequality constraints. These constraints are mainly geometrical constraints.

Figure 3 shows the time evolution of the standard deviations of the sample distribution in the principle axes found by CMA-ES in a typical single trial on test case 1. The overall step-size is disregarded. The picture shows a comparatively ill-conditioned problem with a final condition number of close to a million (the ratio between largest and smallest standard deviation squared). The time evolution of the parameters (not shown) suggests that the problem has also considerable dependencies between the different parameters. This makes the problem difficult to solve.

C. Second Test case

The second test case proposes optimizing the same launcher type with the trajectory module. In this case, the command law parameters are optimized together with the architecture, which gives 23 optimization parameters. Trajectory constraints are added corresponding to the final orbit equality constraints and intermediate inequality constraints such as the maximum heat flux after fairing jettison, the maximum acceleration…

The standard deviation of the sample distribution after 10^5 evaluations shows an even more ill-conditioned problem with a final condition number close to 10^{10}. CMA-ES is particularly well-adapted to deal with such problems.

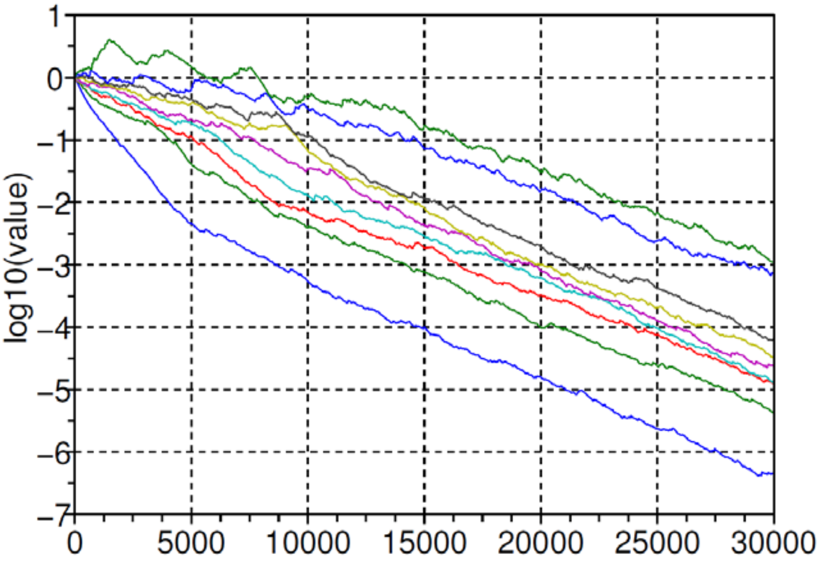

Figure 3 : Standard deviations in the principle axes found by CMA-ES versus number of function evaluations, where the step-size is disregarded

The optimality of the command law obtained is cross-checked with TRAJ1 trajectory optimization tool and a reduced gradient method and it confirmed the good convergence of the CMA-ES algorithm on the command law. The first results show that the CMA-ES provides a good solution to the “all-at-once” problem; optimal architecture and command law are solved together.

	2nd Test case	CMA-ES Optimal solution	Virtual A5 reference
Boosters	Propellant mass (each)	272 tons	236 tons
	Mass Flow-rate (each)	1930 kg/s	1830 kg/s
	Diameter (each)	3.0 m	3.1 m
Main Stage	Propellant mass	140 tons	170 tons
	Mass Flow-rate	225 kg/s	320 kg/s
	Diameter	4.0 m	5.4 m
Upper Stage	Propellant mass	13 tons	17 tons
	Mass Flow-rate	14 kg/s	25 kg/s
	Diameter	3.0 m	5.4 m
Launcher	Cost	ref – 10 %	ref

Table 2 : 2nd test case optimal solution

Table 2 provides the optimal solution of the second test case and a comparison to an advanced project studied in the nineties of Ariane 5 evolution with bigger storable propellant upper stage engine than the one existing on A5ES version[15] of Ariane 5. The results show that we should increase the size of the boosters and reduce the size of the other stages. The algorithm converges towards very thin and long stages that could be problematic from dynamical point of view. The cost is reduced by 10% compared to the reference case. The final solution obtained strongly depends on the model of cost implemented (α and β coefficients).

V. Algorithm performance

To accelerate the convergence process on its MDO platform, Astrium has developed over the year the "split method"[4] where the architecture part of the problem (mass flow rate, propellant mass, diameter) are optimized separately from the trajectory part. The optimization bi-level implementation depicted in Figure 4 was possible with the use of TRAJ2 tool and its fast semi-analytical integration process. The trajectory sub-problem is solved with a reduced gradient method to find the maximum payload. In order to improve the time-consumption, an automatic initialization process was added to find more quickly the optimal command law. Furthermore, a robust error code management system was implemented in the software. The implementation was proved to be successful with the use of NSGA-II alogrithm[16] on mono and multi-criteria optimization.

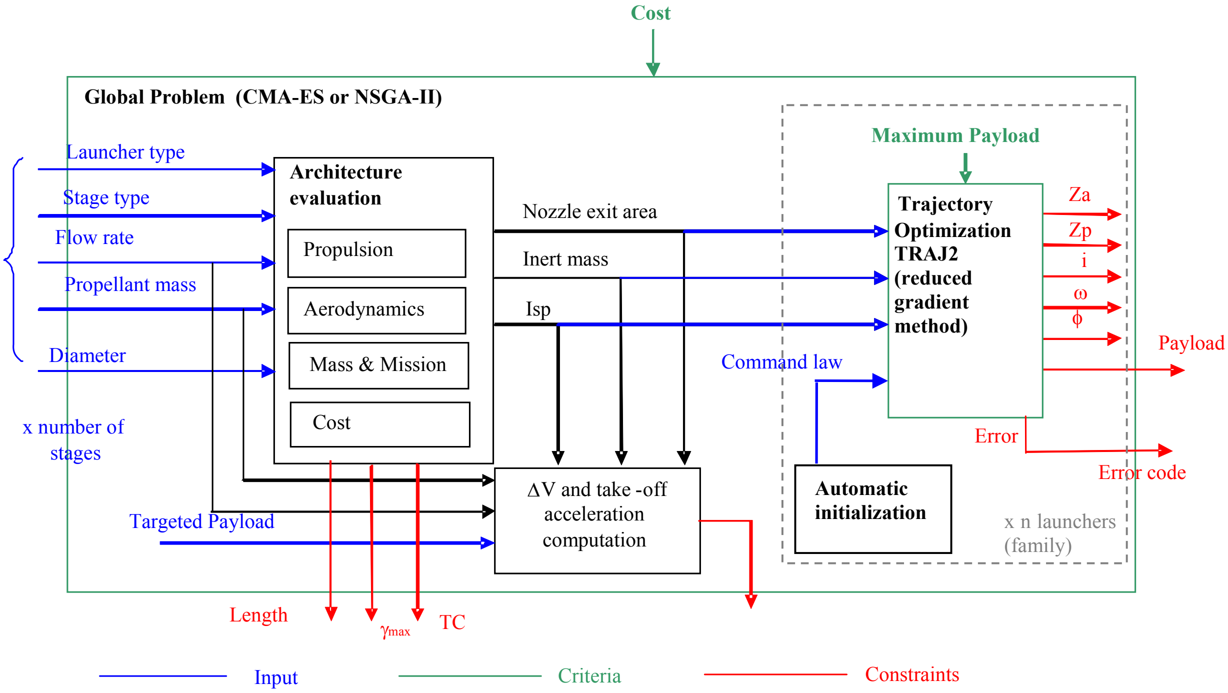

Figure 4 : 'Split Method' implementation

CMA-ES is tested on the architecture part of the "split method", which is equivalent to the first test case (see IV.B) in terms of parameters (9) and constraints except that the velocity increment constraint is replaced by a constraint on the payload. The convergence is good and the optimized parameters are equivalent (1% difference) at each run. The results are compared with the one obtained with non-dominated sorting genetic algorithm NSGA-II and they are very close. These results with single objective optimization confirm the results in multi objective[17].

The results presented in Table 3 were generated with the same test case than in IV.A apart from the target GTO payload set to 6 tons and the upper stage technology cryogenic engine using liquid hydrogen and liquid oxygen instead of storable N_2O_4/MMH propellants. The cost model coefficients have been also changed. The modification of the cost model changes a lot the sharing between the different stages. The optimal launcher strongly depends on the chosen cost model.

		NSGA-II	CMA-ES
Boosters	Propellant mass (each)	75 tons	80 tons
	Mass Flow-rate (each)	975 kg/s	960 kg/s
	Diameter (each)	1.95 m	2.00 m
Main Stage	Propellant mass	125 tons	110 tons
	Mass Flow-rate	315 kg/s	300 kg/s
	Diameter	4.9 m	4.7 m
Upper Stage	Propellant mass	13 tons	13 tons
	Mass Flow-rate	10 kg/s	10 kg/s
	Diameter	3.3 m	3,1 m
Launcher	Cost	~ ref	ref

Table 3 : Comparison between NSGA-II and CMA-ES

VI. Launcher Family Optimization

A. Launcher Family Concept

The concept of launcher family was developed with the aim of reducing development and recurrent launcher cost. The main idea is to use the different stages of a launcher to build other launcher configuration and by this way, to offer a wider payload range to the customer (for instance, adding solid boosters to a linear launcher to increase performance).

The MDO platform with "split method" was completed in order to run several times the trajectory optimizer with different configurations of the launcher to optimize the cost of the whole family. As represented in Figure 4, the grey dashed box is multiplied by the number of launcher configurations in the family. The different bricks (stages, flight sequential…) of the launchers are organized in different ways to fit the definition of the family configurations.

The interest of using MDO concept on a launcher family is to provide a global optimum with respect to the family life profile cost and not single optimum launcher separately. A major difficulty of a launcher family optimization is the error code management. Two or three trajectory optimizations multiply the possible errors for one case.

B. Launcher Family test case

The CMA-ES is tested on a family test case with two launchers and then three launchers. The reference case is derived from a Delta-IV like example. The main launcher has two cryogenic stages (Oxygen – Hydrogen) with two solid boosters. The first derived launcher is common core; the main stage is also used as booster instead of the solid booster. The second derived launcher is the two stage launcher without booster. This test case is only representative of a Delta-IV-like family, most of data are extracted from Delta IV launcher manual[18]. The reference launcher family is close to a family composed of Delta IV M+, Delta IV Heavy and Delta IV Medium launchers but differs from real Delta IV family in the way that Delta IV M+ and Delta IV heavy have different upper stages whereas in our case the same upper stage is used for all 3 launchers. The targeted payloads were set to the results of TRAJ2 with Delta IV data and are a little bit different form the launcher manual payloads. The Delta-IV-like reference is only indicative and provides a point of comparison to the solution obtained after optimization.

The results on a family with 2 launchers with CMA-ES are conclusive compared to the one with NSGA-II. Even if the algorithm does not always converge towards the same exact optimum, the results obtained with CMA-ES are better than the one with NSGA-II. From the same initial guess, CMA-ES always finds a better solution than NSGA-II.

For a family with 3 launchers, the optimization is harder because of numerous error cases. The population has to be increased from 10 to 20 individuals (=launcher families) in CMA-ES. The algorithm succeeds in improving the reference as presented in Table 4. The initial guess of CMA-ES algorithm was provided randomly and the final optimal solution comes very close to the reference case with an improvement of the cost by 4% with a lighter structure on upper stage and smaller boosters. Figure 5 presents the design of the family as obtained after optimization.

		CMA-ES Optimal solution	Delta IV - like reference
Boosters	Propellant mass (each)	20 tons	30 tons
	Mass Flow-rate (each)	275 kg/s	333 kg/s
	Diameter (each)	1,6 m	1,5 m
Main Stage	Propellant mass	195 tons	200 tons
	Mass Flow-rate	995 kg/s	1000 kg/s
	Diameter	5,3 m	5,1 m
Upper Stage	Propellant mass	20 tons	20 tons
	Mass Flow-rate	20 kg/s	24 kg/s
	Diameter	3,4 m	4,1 m
Family	Cost	ref – 4 %	ref

Table 4 : Optimal 3 launcher family

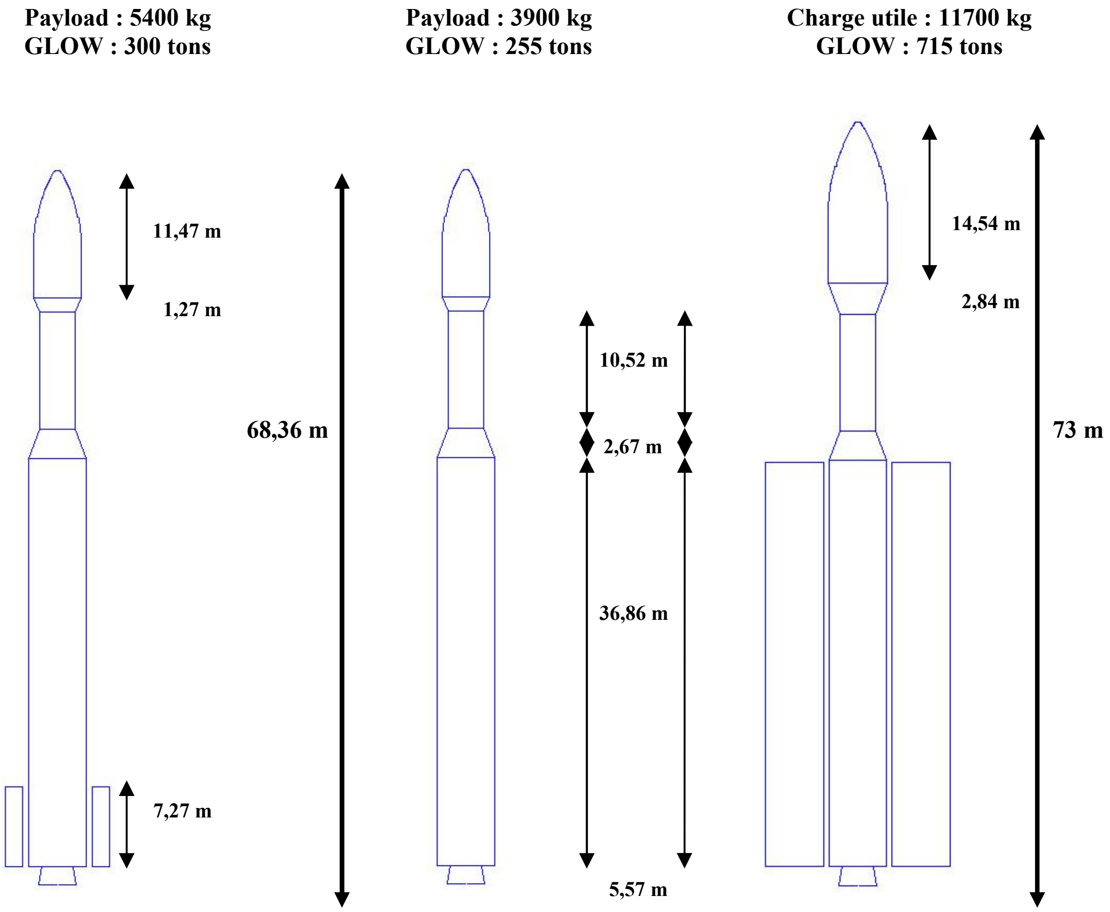

Figure 5 : Optimal 3 launcher family design

VII. Conclusion and Perspectives

As a conclusion, CMA-ES proved to be very powerful for the multidisciplinary optimization of the expendable launch vehicles. The results were as conclusive on “all-at-once” optimization as on launcher family optimization. The coupled trajectory and architecture optimization problem proved to be very ill-conditioned and CMA-ES is particularly well-adapted to deal with it. In the launcher family optimization, CMA-ES was able to cope with numerous error cases and still find a good solution improving the reference case.

CMA-ES performance will certainly permit to add more parameters in the optimization. The main interesting feature to add in the MDO platform would be the multi-point thrust law with the optimization of each point separately. The thrust law profile optimization has strong interactions with trajectory and architecture optimization and should be integrated in multidisciplinary optimization process in particular in the case of solid propulsion. By analyzing some optimal solutions found by the optimizer, a dynamical analysis module could still improve the final design obtained.

References

[1] Jean-Marius, T., “Multidisciplinary Optimization for Early System Design for Expandable Launchers”, *11th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, AIAA-2006-7125, Sept. 2006

[2] Durанté, N., Dufour, A., Pain, V., Baudrillard, G., and Schoenauer, M., “Multidisciplinary Analysis and Optimization Approach for the Design of Expendable Launchers”, *10th AIAA/ISSMO Multidisciplinary Analysis Optimization (MAO) Conference*, AIAA-2004-2441, Sept. 2004

[3] Delattre, N., and Mongrard, O., “The Use of Multi-Disciplinary Optimization for the Design of Expendable Launchers”, *2nd EUCASS*, July 2007

[4] Collange, G., Delattre, N., Lemaire, F., and Quinquis, I., “Different Approaches to Multidisciplinary Optimization For Preliminary Design of Expendable Launch Vehicles”, *3rd EUCASS*, July 2009

[5] Braun, R.D., Moore, A.A., and Kroo, I.M., “Collaborative Approach to Launch Vehicle Design,” *Journal of Spacecraft and Rockets*, Vol. 34, No. 4, 1997, pp. 478-486.

[6] Brown, N., and Olds, J., “Evaluation of Multidisciplinary Optimization (MDO) Technique Applied to a Reusable Launch Vehicle,” AIAA 2005-707, Jan. 2005.

[7] Bayley, D., and Hartfield, R., “Design Optimization of Space Launch Vehicles for Minimum Cost Using a Genetic Algorithm”, AIAA 2007-5852, July 2007

[8] Delattre, N., “Generic all purpose ASTRIUM Space Transportation reference trajectory optimization tools”, in proceedings of the 3rd International Workshop on Astrodynamics tools and Techniques, Oct. 2006

[9] Beyer, H.-G., and Schwefel:, H.-P., “Evolution Strategies: A Comprehensive Introduction”. *Natural Computing* 1(1), 3–52, 2002

[10] Hansen, N., and Ostermeier, A., “Completely Derandomized Self-Adaptation in Evolution Strategies”. *Evol. Comput.* 9(2), 159–195, 2001

[11] Hansen, N., and Kern, S., “Evaluating the CMA evolution strategy on multimodal test functions”. In Xin Yao et al., editor, *Parallel Problem Solving from Nature - PPSN* VIII, pp. 282–291. Springer, 2004.

[12] Auger, A. and Hansen, N. (2005). “A Restart CMA Evolution Strategy With Increasing Population Size”. In Proceedings of the *IEEE Congress on Evolutionary Computation*, CEC 2005, pp.1769-1776

[13] Hansen, N. (2009). “Benchmarking a BI-Population CMA-ES on the BBOB-2009 Function Testbed”. Workshop Proceedings of the *GECCO Genetic and Evolutionary Computation Conference*, ACM, pp. 2389-2395

[14] Collange, G., Delattre, N., Hansen, N., Quinquis, I., and Schoenauer, M., “Multidisciplinary Optimisation in the Design of Future Space Launchers”. In Breitkopf, P., and Coelho, R.-F., eds.: *Multidisciplinary Design Optimization in Computational Mechanics*, Wiley, 2010, pp.487-496;

[15] ”Ariane 5 Users’s Manual Issue 5 Revision 0”, Arianespace, www.arianespace.com, July 2008

[16] Lemaire, F., Collange, G., and Reynaud, S., “GOAL: an industrial MDO platform used to design expendable launchers*”, 3rd International Conference on Multidisciplinary Design Optimization and Applications*, June 2010

[17] C. Igel, N. Hansen, and S. Roth. “Covariance matrix adaptation for multi-objective optimization”. *Evolutionary Computation*, 15(1) p1-28, 2007.

[18] “Delta IV Payload Planners Guide”, United Launch Alliance, www.ulalaunch.com , September 2007