TESTING APPROACHES FOR GLOBAL OPTIMIZATION OF SPACE TRAJECTORIES

Massimiliano Vasile and Edmondo Minisci
Department of Aerospace Engineering, University of Glasgow
James Watt South Building, G12 8QQ, Glasgow, UK
mvasile,eminisci@eng.gla.ac.uk

Marco Locatelli
Dipartimento di Informatica, Università di Torino
Corso Svizzera 185, 10149 Torino, Italy
locatell@di.unito.it

Abstract In this paper the procedures to test global search algorithms applied to space trajectory design problems are discussed. Furthermore, a number of performance indexes that can be used to evaluate the effectiveness of the tested algorithms are presented. The performance indexes are then compared and the actual significance of each one of them is highlighted. Three global optimization algorithms are tested on three typical space trajectory design problems.

Keywords: Global optimization, Stochastic optimization, Space trajectory design.

1. Introduction

In the last decade many authors have used global optimization techniques to find optimal solutions to space trajectory design problems. Many different methods have been proposed and tested on a variety of cases. From pure Genetic Algorithms (GAs) [7, 5, 8, 1] to Evolutionary Strategies (ESs) (such as Differential Evolution, DE)[12] to hybrid methods [14], the general intent is to improve over the pure grid or enumerative search. Sometimes, the actual advantage of using a global method is difficult to appreciate, in particular when stochastic based techniques are used. In fact, if, on one hand, a stochastic search provides a non-zero probability to find an optimal solution even with a small number of function evaluations, on the other hand, the repeatability of the result and therefore the reliability of the method can be questionable. The first actual assessment of the suitability of global optimization methods to the solution of space trajectory design problems can be found in two studies by the University of Reading [6] and by the University of Glasgow [11]. One of the interesting outcomes of both studies was that DE performed particularly well on most of the problems, compared to other methods. In both studies, the indexes of performance for stochastic methods were: the average value of the best solution found for each run over a number of independent runs, the corresponding variance and the best value from all the runs. For deterministic methods, the index of performance was the best value for a given number of function evaluations. In this paper, we propose a testing methodology for global optimization methods addressing specifically black-box problems in space trajectory design. In particular, we focus our attention on stochastic based approaches. The paper discusses the actual significance of a number of performance indexes and proposes an approach to test a global optimization algorithm.

2. Testing Procedure

In this section we describe a testing procedure that can be used to investigate the complexity of the problem and to derive the performance indexes described in the next section. If we call A a generic solution algorithm and p a generic problem with objective function f, we can define a general procedure as in Algorithm 1.

Algorithm 1 Convergence Test

set the max number of function evaluations for A equal to N
apply A to p for n times
for all $i \in [1, ..., n]$ **do** $\phi(N, i) = \min f(A(N), p, i)$
end for
compute: $\phi_{min}(N) = \min_{i \in [1,...,n]} \phi(N, i)$, $\phi_{max}(N) = \max_{i \in [1,...,n]} \phi(N, i)$

Now if the algorithm A is convergent, when the number of function evaluations N goes to infinity the two functions ϕ_{min} and ϕ_{max} converge to the same value, the global minimum value denoted as f_{global}. If we fix a tolerance value tol_f, we could consider the following random variable as a possible quality measure of an algorithm

$$N^* = \min\{\phi_{max}(N) - f_{global} \leq tol_f \;:\; \forall\, N \geq N^*\}.$$

The larger (the expected value of) N^* is, the slower is the convergence of A. However, such measure can be unpractical since, though finite,

N^* could be very large. In practice, what we would like is not to choose N large enough so that a success is always guaranteed, but rather, for a fixed N value, we would like to maximize the probability of hitting a global minimizer. Now, let us define the following quantity:

$$\delta_f(\mathbf{x}) = f(\mathbf{x}) - f_{global} \tag{1}$$

(in case the global minimum value f_{global} is not known, we can substitute it with the best known value f_{best}). We can now define a new procedure, summarized in Algorithm 2.

Algorithm 2 Convergence to the global optimum

set the max number of function evaluations for A equal to N
apply A to p for n times
set $j = 0$
for all $i \in [1, ..., n]$ **do**
$\phi(N, i) = \min f(A(N), p, i)$
$\mathbf{x} = arg\ \phi(N, i)$
compute $\delta_f(\mathbf{x})$
  if $(\delta_f(\mathbf{x}) < tol_f)$ **then** $j = j + 1$
  end if
end for

The output of such procedure is the fraction j/n of the n runs of A which end up with a success, i.e. an estimate of the probability of success for A. A key point is properly setting the value of n, because a value of n too small would correspond to an insufficient number of samples to have a proper statistics. This choice will be discussed in Section 2.1. The value of the tolerance parameter tol_f, which defines the concept of success, is problem dependent. Note that, in the case of multiple minima with equal f also the distance $\|x - x_{global(best)}\|$ would be relevant, however in the following we are only interested in the value of the merit function.

2.1 Performance Indexes

Now that the testing procedure is defined we can define the performance indexes. For a stochastic based algorithm different performance indexes can be defined. In the following we will discuss about the significance of some of them keeping in mind the practical use of a global optimization, or global search, algorithm in space trajectory design.

The current practice is mainly focused on the evaluation of best value, mean and variance values of the best solutions found on n runs [8, 7, 12].

An algorithm is considered as better performing as the obtained mean value is closer to the global optimum and a small variance is considered as a suggestion of robustness. This approach, however, does not consider three main issues: a) generally the distribution of the best values cannot be approximated with a gaussian distribution; b) from a practical standpoint, we are not interested in the mean values, which could be faraway from the global optimum; c) we want to know the level of confidence in the repetibility and global optimality of the results.

An alternative index that can be used to assess the effectiveness of a stochastic algorithm is the success rate, which is related to the j value in Algorithm 2, being $Sp = j/n$. Considering the success as the referring index for a comparative assessment implies two main advantages. First, it gives an immediate and unique indication of the algorithm effectiveness, addressing all the issues highlighted above, and, second, the success rate can be represented with a binomial probability density function (PDF), independently of the number of function evaluations, the problem and the type of optimization algorithm. This latter means, moreover, that we can design the experiment and fix the number of runs, n, on the basis of the error we can accept on the success value. A usual starting point to sample size determination for a binomial distribution is to assume both the normal approximation for the sample proportion p of successes, i.e. $p \sim N\{\theta, \theta(1-\theta)/n\}$, and the requirement that $Pr[|p-\theta| \leq d|\theta]$ should be at least $1-\alpha$ [2]. This leads to expression in (2) and to the conservative rule in (3), obtained if $\theta = 0.5$

$$n \geq \theta(1-\theta)\chi^2_{(1),\alpha}/d^2 \tag{2}$$

$$n \geq 0.25\chi^2_{(1),\alpha}/d^2 \tag{3}$$

For our tests we considered $n = 200$, which should “guarantee” an error ≤ 0.05 ($d = 0.05$) with a 95% confidence ($\alpha = 0.05$).

3. Problem Description

Three different test-cases, with different difficulty levels, are considered. In all of these cases the objective will be to minimize the variation of the velocity of the spacecraft due to a propelled maneuver, Δv. Minimizing the Δv means minimizing the propellant mass required to perform the maneuver, since propellant mass increases exponentially with Δv.

A simple, but already significant, application is to find the best launch date and time of flight to transfer a spacecraft from Earth to the asteroid Apophis. The transfer is computed as the solution of a Lambert's problem [3], therefore the design variables are the departure date from the first celestial body, t_0 and the flight time T_1 from the first to the second body. The launch date from the Earth has been taken in the interval [3653, 10958] (number of elapsed days since January 1st 2000, MJD2000), while the time of flight has been taken in the interval [50, 900] days. The best known solution is f_{best}=4.3745658 km/s.

The second test-case consists of a transfer from Earth to Mars with the use of the relative movement and gravity of Venus to alter the path and speed of the spacecraft in order to save fuel. The mission is implemented as two Lambert arcs, Earth-Venus and Venus-Mars, plus a gravity assist maneuver at Venus. The problem has dimension 6, t_0 [d, MJD2000] [3650, 3650+365.25*15], T_1 [d] [50, 400], γ_1 [rad] [$-\pi$, π], $r_{p,1}$ [1, 5], α_2 [0.01, 0.9], T_2 [d] [50, 700], where γ_1, $r_{p,1}$ and α_2 are related to the gravity assist maneuver and are the angle of hyperbola plane, the radius of the pericentre of the hyperbola adimensionalised with the radius of the planet, and the fraction of time of flight before the deep space manoeuvre, respectively. The best known solution is f_{best}=2.9811 km/s. The third test is a multi gravity assist trajectory from the Earth to Saturn following the sequence Earth-Venus-Venus-Earth-Jupiter-Saturn (EVVEJS). Gravity assist maneuvers have been modeled through a linked-conic approximation with powered maneuvers, i.e., the mismatch in the outgoing velocity is compensated through a Δv maneuver at the GA planet. No deep-space maneuvers are possible and each planet-to-planet transfer is computed as the solution of a Lambert's problem. The objective function is given in [9] and also in this case the dimensionality of the problem is 6, t_0 [d, MJD2000] [-1000, 0], T_1 [d] [30, 400], T_2 [d] [100, 470], T_3 [d] [30, 400], T_4 [d] [400, 2000], T_5 [d] [1000, 6000]. The best known solution is f_{best}=4.9307 km/s. Due to format requirements, it is not possible to exhaustively describe the problems, but they are freely available on request as black-box executables.

4. Used Algorithms

We tested three global search algorithms belonging to the class of stochastic algorithms. More precisely, one belongs to the class of ESs, one to the class of GAs and one to the class of agent-based algorithms.

We considered 6 different settings for the DE, resulting from combining 3 sets of populations, $[5\,d, 10\,d, 20\,d]$, where d is the dimensionality of the problem, 2 strategies, 6 (DE, best, 1, bin) and 7 (DE, rand, 1, bin) [13], and single values of step-size and crossover probability, $F = 0.75$ and $CR = 0.8$ respectively, on the basis of common use. DEs with strategy 6 are indicated, in Table 1, as Algorithms 1, 2 and 3, while those with strategy 7 are 4, 5 and 6, depending on the population size (from the smallest to the biggest).

For the Particle Swarm Optimization (PSO) algorithm ([4]), 9 different settings were considered, resulting from the combination of 3 sets of population, again $[5\,d, 10\,d, 20\,d]$, 3 values for the maximum velocity bound, $V_{max} \in [0.5, 0.7, 0.9]$ (corresponding in Table 1 to Algorithms 7-9, 10-12 and 13-15, respectively), and single values for weights, $C_1 = 1$ and $C_2 = 2$. Regarding the GA ([10]) application, only the influence of the population size was considered ([100, 200, 400] for the bi-impulse test case and [200, 400, 600] for the other two cases, corresponding to Algorithms 16-18 in Table 1), with single values for crossover and mutation probability, $Cr = 1$ and $Mp = 1/d$. All algorithms operated on normalized ([0,1]) search spaces.

Table 1. Numbering of tested algorithms

Alg.	Id.	Alg.	Id.	Alg.	Id.
DE($5\,d$, 6)	1	PSO($5\,d$, 0.5)	7	PSO($5\,d$, 0.9)	13
DE($10\,d$, 6)	2	PSO($10\,d$, 0.5)	8	PSO($10\,d$, 0.9)	14
DE($20\,d$, 6)	3	PSO($20\,d$, 0.5)	9	PSO($20\,d$, 0.9)	15
DE($5\,d$, 7)	4	PSO($5\,d$, 0.7)	10	GA(100)	16
DE($10\,d$, 7)	5	PSO($10\,d$, 0.7)	11	GA(200)	17
DE($20\,d$, 7)	6	PSO($20\,d$, 0.7)	12	GA(400)	18

5. Comparison Among Performance Indexes

The results of the tests are summarized in Table 2 and 3, where success probability (Table 2) and best value, mean and variance of best results (Table 3) are given for each of 18 (set) solvers. For both EA and EVM cases, success probability allows a fair classification and gives a clear indication of the best performing algorithms. Algorithms 5 and 6 perform undoubtedly much better than the others and GAs (Algorithms 16-18) appear to be the worst performing ones. Algorithm 4 wins a bronze medal, but if we can be confident on its third position for the EVM problem, we cannot have the same level of confidence regarding the third position for EA, because of the proximity of other algorithms. Actually, due to the binomial nature of the success and the adopted sample size, it is not possible to fairly discriminate between algorithms for which the success distance is smaller than the expected error (0.05 in our computations). Therefore, Algorithm 4 has to be considered at the same level of Algorithms 11, 13 and other PSO settings. For the same reasons, we can say that, among the PSO settings, Algorithms 11 and

13 perform better than Algorithm 8 but the remaining PSO algorithms work at the same level.

Table 2. Success for the 18 algorithms on the three test-cases. To compute the success, following tol_f values were used: 0.001 for EA, $3 - f_{best}$ for EVM and $5 - f_{best}$ for EVVEJS

	1	2	3	4	5	6	7	8	9
EA	0.140	0.300	0.355	0.450	0.770	0.855	0.355	0.345	0.410
EVM	0.050	0.050	0.050	0.150	0.250	0.370	0.040	0.035	0.080
EVVEJS	0.020	0.005	0.015	0.000	0.000	0.000	0.000	0.005	0.000
	10	11	12	13	14	15	16	17	18
EA	0.395	0.425	0.410	0.435	0.385	0.420	0.160	0.240	0.105
EVM	0.045	0.060	0.055	0.035	0.070	0.075	0.005	0.010	0.035
EVVEJS	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.005

An analogous vagueness is associated to the results of a number of algorithms when applied to the EVM case and to almost all of them when applied to the EVVEJS case. For the EVVEJS transfer, in particular, the probability of success, though not always 0, is at best, including the error margin, not greater than 0.07.

Table 3. Indexes: Best value, Mean Best, Variance Best.

	EA (N=5000)			EVM (N=100000)			EVVEJS (N=400000)		
1	4.3746	4.6962	0.0736	2.9811	3.6032	0.3240	4.9307	12.5129	15.0723
2	4.3746	4.5734	0.0312	2.9811	3.5118	0.0880	4.9307	11.3672	15.7534
3	4.3746	4.5198	0.0166	2.9811	3.4335	0.0823	4.9307	9.9694	15.9349
4	4.3746	4.5126	0.0236	2.9811	3.2936	0.0307	5.3034	8.1468	9.7486
5	4.3746	4.4197	0.0074	2.9811	3.2336	0.0277	5.3034	6.3851	5.0131
6	4.3746	4.3919	0.0026	2.9813	3.1699	0.0285	5.3034	5.5596	1.3999
7	4.3746	4.5120	0.0124	2.9811	3.8194	0.6794	5.0275	12.6827	16.4910
8	4.3746	4.5103	0.0119	2.9811	3.7812	0.6252	4.9558	11.9736	18.7031
9	4.3746	4.4990	0.0125	2.9811	3.6537	0.5233	5.3034	11.1931	17.9212
10	4.3746	4.5098	0.0142	2.9811	4.0427	1.0135	5.0125	11.7365	17.7327
11	4.3746	4.4919	0.0120	2.9811	3.9285	0.8346	5.0553	10.7274	17.2695
12	4.3746	4.5037	0.0136	2.9811	3.7329	0.5971	5.0177	10.4668	18.4276
13	4.3746	4.4959	0.0133	2.9811	4.2185	1.0569	5.2450	11.8344	21.7106
14	4.3746	4.5503	0.3720	2.9811	3.9747	0.8676	5.0223	10.5636	18.4262
15	4.3746	4.4983	0.0131	2.9811	3.8127	0.7466	5.0310	10.5256	15.1327
16	4.3746	4.5743	0.0260	2.9885	3.7821	0.2413	5.1595	10.6525	15.1862
17	4.3746	4.4959	0.0146	2.9926	3.5435	0.1400	5.0242	8.3140	9.9140
18	4.3746	4.4507	0.0084	2.9827	3.4452	0.0983	4.9821	6.9770	6.6833

For cases when the success probability cannot give practically useful information to classify the algorithms, the user could be tempted to use mean and variance values, but this practice is strongly heedless. Since, as anticipated in Section 2.1 and confirmed by tests (see Figure 1), the PDF of the best values is not a gaussian and, moreover, changes during the process, mean and variance values are not enough to understand the algorithm behaviour and we cannot say anything about their exactness.

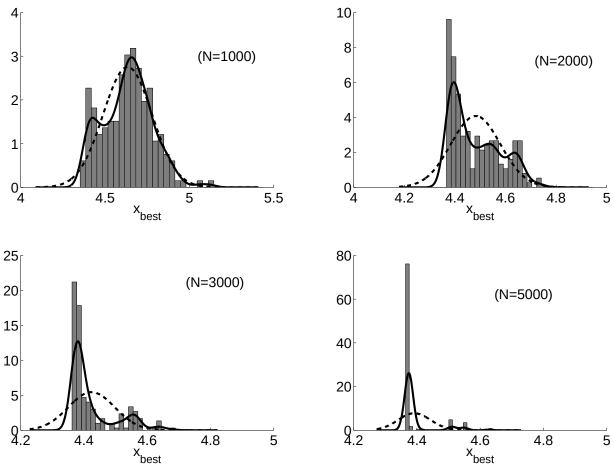
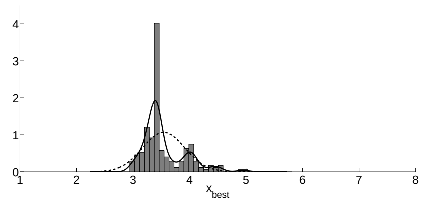


Figure 1. Variation of the PDF for best solutions with $N$for the Algorithm 6 applied to EA test-case; discrete, incorrect gaussian approximation (dashed) and kernel based approximation (continuous) are shown.

Even if we suppose mean and variance are correct (in some way), looking at these two values can bring to incorrect conclusions. For instance, if we consider the values for the Algorithms 14 and 17 applied to EVM, we could conclude that Algorithm 17 performs better than Algorithm 14, because of a smaller mean value and a smaller variance (regarded as an index of robustness). But if we are interested in global optimal solutions, Algorithm 14 is noticeably better: it is able to find the global solution, even if it is less robust and gets stuck many times in a far basin (see Figure 2).

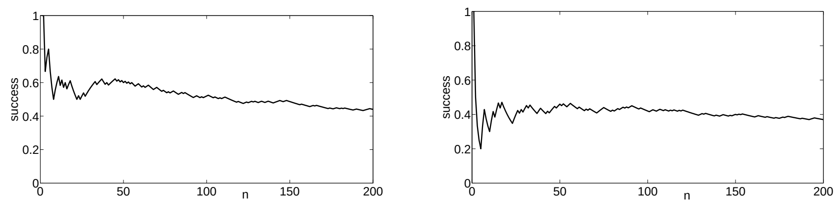
In order to solve an uncertainty condition, for instance when the success probability appears uniformly null, relaxing the tol_f value could be useful. Focusing on the EVVEJS case, there is no way to correctly discriminate among the algorithms on the basis of data in Table 2, but if the success threshold is raised from 5 to 5.3, then a superior performance of GAs is revealed. Most likely, this behaviour is due to a combination of large population size, mutation operator and non-deterministic selection, which reduce the local convergence, and allows for a better exploration of the search space.



(a) GA (b) PSO

Figure 2. Examples of two PDFs, which could bring to incorrect conclusions; for both cases, discrete, incorrect gaussian approximation (dashed) and kernel based approximation (continuous) are shown.

As previously stated, for all the tests, 200 runs were performed in order to maintain the error on the success probability within a predefined margin. The extreme importance of the sample size appears evident when we look at Figure 3, where the variation of the success probability is shown as function of n. For $n \leq 50$, the success is extremely oscillating and the confidence on the obtained value should be considered poor.



(a) Alg. 13 on EA case (b) Alg. 6 on EVM case

Figure 3. The influence of sample size. The success probability is shown as function of the sample size for two different algorithm/test-case combinations.

6. Conclusions

The work focuses on the testing procedures for the application of global optimization algorithms to space trajectory design and tries to set the basis for a standard and consistent procedure. The current testing practice and the currently used performance indexes are criticized and a preliminary testing/analysis procedure is proposed and the probability of success is indicated as the most useful index, when performance of different algorithms are to be compared. Moreover, the binomial nature of this index allows to link the number of performed runs to the expected error on the success itself, while it is not possible to have the same statistical consistency when mean and variance values are utilized, because of the unknown nature of the PDF for the best values. In general, it should be stressed that if the comparative tests have to be reliable, the number of runs cannot be lower than a threshold depending on the nature of the considered indexes.

In the future, the testing procedure will be improved by considering also the heuristics costs and the link between the performance of some heuristics and the main structures of the test-cases.

References

[1] O. Abdelkhalik and D. Mortari. N-Impulse Orbit Transfer Using Genetic Algorithms. *Journal of Spacecraft and Rockets*, 44(2):456–459, 2007.

[2] C.J. Adcock. Sample size determination: a review. *The Statistician*, 46(2):261–283, 1997.

[3] H. Battin. *An Introduction to the Mathematics and Methods of Astrodynamics*. AIAA, 1999.

[4] M. Clerc. *Particle Swarm Optimization*. ISTE, 2006.

[5] V. Coverstone-Carroll. Near-optimal low-thrust orbit transfers generated by a genetic algorithm. *Journal of Spacecraft and Rockets*, 33(6):859–862, 1996.

[6] P. Di Lizia and G. Radice. Advanced Global Optimization Tools for Mission Analysis and Design. ESA Ariadna Report ITT AO4532/18139/04/NL/MV, Call 03/4101, 2004.

[7] P.J. Gage, R.D. Braun and I.M. Kroo. Interplanetary trajectory optimization using a genetic algorithm. *J. of the Astronautical Sciences*, 43(1):59–75, 1995.

[8] Y.H. Kim and D.B. Spencer. Optimal Spacecraft Rendezvous Using Genetic Algorithms. *Journal of Spacecraft and Rockets*, 39(6):859–865, 2002.

[9] A.V. Labunsky, O.V. Papkov, K.G. Sukhanov. *Multiple Gravity Assist Interplanetary Trajectories*. ESI Book Series, 1998.

[10] M. Mitchell. *An introduction to genetic algorithms*. MIT Press, 1998.

[11] D.R. Myatt, V.M. Becerra, S.J. Nasuto, and J.M. Bishop. Advanced Global Optimization Tools for Mission Analysis and Design. ESA Ariadna Report ITT AO4532/18138/04/NL/MV, Call03/4101, 2004.

[12] A.D. Olds, C.A. Kluever and M.L. Cupples. Interplanetary Mission Design Using Differential Evolution. *Journal of Spacecraft and Rockets*, 44(5):1060–1070, 2007.

[13] K.V. Price, R.M. Storn and J.A. Lampinen. *Differential Evolution. A Practical Approach to Global Optimization*. Natural Computing Series, Springer, 2005.

[14] M. Vasile, L. Summerer and P. De Pascale. Design of Earth-Mars Transfer Trajectories using Evolutionary Branching Techniques. *Acta Astronautica*, 56:705–720, 2005.