Benchmarking different global optimisation techniques for preliminary space trajectory design

Tamás Vinkó[†] and Dario Izzo[†] and Claudio Bombardelli[†]

[†]ESA, Advanced Concepts Team, ESTEC Keplerlaan 1, Postbus 299,2200 AG, Noordwijk

contact: tamas.vinko@esa.int, dario.izzo@esa.int

Abstract

In this paper we describe a number of global optimisation problems connected to spacecraft trajectory design. Each problem is coded in the form of a blackbox objective function accepting, as inputs, the decision vector and returning the objective function and the constraint evaluation. The code is made available on line as a challenge to the community to develop performing algorithms able to solve each of the problems proposed in an efficient manners. All the problems proposed draw inspiration from real trajectory problems, ranging from Cassini to Rossetta to Messenger to possible future missions. As a start we also report the results coming from applying standard global optimisation algorithm to each of the problem. We consider Differential Evolution, Particle Swarm Optimisation, Genetic Algorithm, Adaptive Simulated Annealing and GLOBAL. As all these standard implementations seem to fail to solve more complex problems, we conclude the paper suggesting a cooperative approach between the different algorithm showing performance improvements.

Introduction

Spacecraft trajectory design problems can be formalized as optimisation problems. Multiple gravity assist (MGA) missions, with the possibility of using deep space manouvre (MGADSM), are essentially global optimisation problems, which is, generally speaking, the task of finding the absolutely best value of a nonlinear function under given constraints. Good solutions to these problems can also serve as starting points for low-thrust trajectory optimisation, usually approached with local optimisation methods. Recently, many papers are dealing with global trajectory optimisation [1, 2, 3, 4, 5, 6], however, there is no standard way to state how relevant the achieved result is. The aim of this paper is to propose a collection of interesting and realistic test problems as challenging benchmark problems together with test results obtained by systematic run of some well-known and widely used global optimisers. We applied our collaborative distributed global optimiser [7] in order to get results which can be used as references. These results are an invitation the communities – to the global optimisation community and to the aerospace engineers–, to develop, apply and compare their own algorithms on these problems. All the proposed models are coded in Matlab and C/C++ and made available to download on our website `www.esa.int/gsp/ACT/inf/op/globopt.htm`. References and weblinks to the solvers used in this paper can also be found in this location. Should any research group find better solutions than the ones reported in the web site and in this paper, they are encouraged to submit them to us (act@esa.int) with a short description of the method, so that we can update the web site.

The models

In this paper we essentially present two types of trajectory models we will refer to as the MGA problem and the MGADSM problem. The first type of problem, described in details in [3], represents an interplanetary trajectory of a spacecraft equipped with chemical propulsion and able to thrust only during its planetocentric phases. This simple model is quite useful in a number of preliminary trajectory calculations and has the advantage of resulting in a small dimensional optimisation problem that has been proven to be suitable for a pruning process having polynomial complexity both in time and in space and that results into an efficient computer implementation (GASP [3]) . On the other hand, the constraint on the spacecraft thrusting only during the planetocentric hyperbolae is often unacceptable as it may results in trajectories that are not realistic or that use more propellant than necessary. A more complete problem is the MGADSM. This represent again an interplanetary trajectory of a spacecraft equipped with chemical propulsion, able to thrust its engine once at any time between each trajectory leg. Thus the solutions to this problem are suitable to perform preliminary quantitative calculation for real space missions. This comes to the price of having to solve an optimisation problem of larger dimensions. The implementation details of this problem are the sum of a number of previously published works [4, 5, 6, 8] and thus are briefly reported. The generic form of the MGADSM problem can be written as:

$$\begin{aligned} \text{find:} \quad & \mathbf{x} \in \mathrm{R}^n \\ \text{to minimise:} \quad & J(\mathbf{x}) \\ \text{subject to:} \quad & \mathbf{g}(\mathbf{x}) \end{aligned}$$

where $\mathbf{x}$ is our decision vector, J is the objective function and $\mathbf{g}$ are non linear constraint that may come from operational considerations or from the spacecraft system design. Given a planetary sequence of N planets, the decision vector is defined by:

$$\begin{aligned} \mathbf{x} = & [t_0, V_\infty, u, v, \eta_1, T_1, \\ & r_{p2}, b_{incl2}, \eta_2, T_2, ... \\ & ..., r_{p_{N-1}}, b_{incl_{N-1}}, \eta_{N-1}, T_{N-1}] \end{aligned}$$

As a consequence a typical MGADSM problem will have dimension $d = 6+(N-2)*4$. In the decision vector, t_0 represent the spacecraft launch date, V_∞, u, v define the heliocentric direction of the departure hyperbolic velocity $\mathbf{v}_\infty$ according to the formulas:

$$\begin{aligned} \theta &= 2\pi u \\ \varphi &= \arccos(2v-1) - \pi/2 \\ \mathbf{v}_\infty / V_\infty &= \cos(\theta)\cos(\varphi)\mathbf{i} + \sin(\theta)\cos(\varphi)\mathbf{j} + \\ & + \sin(\varphi)\mathbf{k} \end{aligned}$$

where the frame $\mathbf{i}, \mathbf{j}, \mathbf{k}$ is defined by

$$\begin{aligned} \mathbf{i} &= \mathbf{v}(t_0) / \parallel \mathbf{v}(t_0) \parallel \\ \mathbf{k} &= \mathbf{r}(t_0) \times \mathbf{v}(t_0) / \parallel \mathbf{r}(t_0) \times \mathbf{v}(t_0) \parallel \\ \mathbf{j} &= \mathbf{z} \times \mathbf{i} \end{aligned}$$

and $\mathbf{r}(t_0), \mathbf{v}(t_0)$ are the heliocentric velocity and position of the departure planet at t_0. Once the spacecraft heliocentric position $\mathbf{r}(t_0)$ and velocity is known $\mathbf{v}_{s/c} = \mathbf{v}(t_0) + \mathbf{v}_\infty$ its trajectory gets propagated along a keplerian orbit for the time $\eta_1 T_1$ and from the arrival point a Lambert problem [9] is solved that brings the spacecraft position to match that of the second planet in sequence in the time $(1-\eta_1)T_1$. If $N > 2$ each subsequent i-th trajectory phase will be determined by first evaluating the fly-by geometry:

$$\begin{aligned} \tilde{\mathbf{v}}_{in} &= \mathbf{v}_{in} - \mathbf{v}_{pla} \\ e &= 1 + r_{p_i}/\mu_{pla} \|\tilde{\mathbf{v}}_{in}\| \\ \delta &= 2 \arcsin 1/e \\ \mathbf{i}_x &= \tilde{\mathbf{v}}_{in} / \|\tilde{\mathbf{v}}_{in}\| \\ \mathbf{i}_y &= \mathbf{i}_x \times \mathbf{r}_{pla} / \|\mathbf{i}_x \times \mathbf{r}_{pla}\| \\ \mathbf{i}_z &= \mathbf{i}_x \times \mathbf{i}_y \\ \tilde{\mathbf{v}}_{out} / \|\tilde{\mathbf{v}}_{in}\| &= \cos\delta \mathbf{i}_x + \sin i_B \sin\delta \mathbf{i}_y + \\ & + \cos i_B \sin\delta \mathbf{i}_z \\ \mathbf{v}_{out} &= \mathbf{v}_{pla} + \tilde{\mathbf{v}}_{out} \end{aligned}$$

where μ_{pla} is the planet gravitational constant. Once the spacecraft velocity $\mathbf{v}_{out}$ is known, we propagate the spacecraft trajectory along a keplerian orbit for the time $\eta_i T_i$. From the arrival point a Lambert problem [9] is then solved to bring the spacecraft position to match that of the $i+1$-th planet in sequence in the time $(1-\eta_i)T_i$.

The objective function $J(\mathbf{x})$ typically measure the propellant consumption of the spacecraft, but can also be related to other objectives such as the total mission time, the spacecraft mass, the properties of the final orbit acquired and so on. Also the non linear constraint functions $\mathbf{g}(\mathbf{x})$ depend on the problem considered and are defined case by case.

The Problems

In this section all the benchmarking problems are listed with their complete parameter description. The first two are MGA problems and then we are describing four MGADSM problems. The best solutions known (the solution vectors together with the corresponding objective values) are reported as well. These were obtained using DiGMO [7], a distributed cooperative global optimisation technique which uses several population based global solvers and able to learn the best solver combination to approach complex problems.

Cassini1 (EVVEJS.mat)

This is an MGA problem that is related to the Cassini spacecraft trajectory design problem (a more complex representation of this problem is found later). The objective of this mission is to reach Saturn and to be captured by its gravity into an orbit having pericenter radius $r_p = 108950$ km, and eccentricity $e = 0.98$. The planetary fly-by sequence considered is Earth-Venus-Venus-Earth-Jupiter-Saturn (as the one used by Cassini spacecraft).

Table 1: State vector bounds in Cassini1.

State	Variable	LB	UB	Units
x(1)	t_0	-1000	0	MJD2000
x(2)	T_1	30	400	days
x(3)	T_2	100	470	days
x(4)	T_3	30	400	days
x(5)	T_4	400	2000	days
x(6)	T_5	1000	6000	days

As objective function we use the total ΔV accumulated during the mission, including the launch ΔV and the various ΔV one needs to give at the planets and upon arrival to perform the final orbit injection. For the six dimensional state vector we use the bounds given in Table 1. As constraints we limit the various fly-by pericenter to the values: $r_{p1} \geq 6351.8$ km, $r_{p2} \geq 6351.8$ km, $r_{p3} \geq 6778.1$ km, $r_{p4} \geq 600000$ km. The best solution known for this problem is

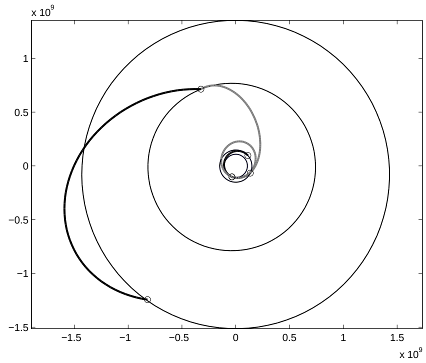

Figure 1: Best solution known for the problem Cassini1.

$\mathbf{x} = [-789.8055, 158.33942, 449.38588, 54.720136, 1024.6563, 4552.7531]$, corresponding to a final objective function of 4.93 km/sec. This solution is shown in Figure 1.

GTOC1 (EVEVEJSA.mat)

This problem draws inspiration from the first edition of the Global Trajectory Optimisation Competition (GTOC1) [10]. More information on the yearly event can be found at `www.esa.int/gsp/ACT/mad/op/GTOC/index.htm`.

It is, again, an MGA problem [3] with a rather long fly-by sequence including mainly Earth and Venus. The final target is the asteroid TW229. The objective of the mission is to maximise the change in sami-major axis of the asteroid orbit following an anaelastic impact of the spacecraft with the asteroid $J(\mathbf{x}) = m_f \mathbf{U} \cdot \mathbf{v}$. As constraints we limit the various fly-by pericenters to the values values: $r_{p1} \geq 6351.8$ km, $r_{p2} \geq 6778.1$ km, $r_{p3} \geq 6351.8$ km, $r_{p4} \geq 6778.1$ km, $r_{p5} \geq 600000$ km, $r_{p6} \geq 70000$ km. We also consider a launcher ΔV of 2.5 km/sec, a specific impulse of $I_{sp} = 2500$s and a spacecraft initial mass of $m_0 = 1500$kg. For the eight dimensional state vector we use the bounds given in Table 2.

The best solution known for this problem is $\mathbf{x} =$ [6809.476683160, 169.598512787, 1079.375156244, 56.53776494142, 1044.014046276, 3824.160968179 1042.885114734 3393.057868710], corresponding to a

Table 2: State vector bounds in GTOC1

State	Variable	LB	UB	Units
x(1)	t_0	3000	10000	MJD2000
x(2)	T_1	14	2000	days
x(3)	T_2	14	2000	days
x(4)	T_3	14	2000	days
x(5)	T_4	14	2000	days
x(6)	T_5	100	9000	days
x(7)	T_6	366	9000	days
x(8)	T_7	300	9000	days

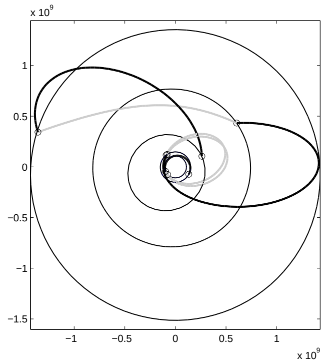

Figure 2: Best solution known for the problem GTOC1.

final objective function of 1,580,599 kg km^2/sec^2. Figure 2 shows this solution. Note that the arc joining Saturn with the asteroid is forced to be retrograde when solving the relative Lambert problem.

SAGAS (EdEdJ.mat)

In this trajectory problem we design what is commonly called a ΔV-EGA manouvre to then fly-by Jupiter and reach 50AU. The objective function considered is the overall mission length and has to be minimised. This creates an MGADSM problem where two more variables need to be added to the decision vector in order to be able to evaluate the keplerian orbit reached after the last fly-by. As constraints we consider the ΔV capability of the spacecraft $\Delta V_1 + \Delta V_2 < 1.782$ and the total available $\Delta V = \Delta V_1 + \Delta V_2 + \Delta V_\infty < 6.782$ km/s. For the twelve dimension state vector we use the bounds given in Table 3.

Table 3: State vector bounds in SAGAS.

State	Variable	LB	UB	Units
x(1)	t_0	7000	9100	MJD2000
x(2)	V_∞	0	7	km/sec
x(3)	u	0	1	n/a
x(4)	v	0	1	n/a
x(5)	T_1	50	2000	days
x(6)	T_2	300	2000	days
x(7)	η_1	0.01	0.9	n/a
x(8)	η_2	0.01	0.9	n/a
x(9)	r_{p1}	1.05	7	n/a
x(10)	r_{p2}	8	500	n/a
x(11)	b_{incl1}	$-\pi$	π	rad
x(12)	b_{incl2}	$-\pi$	π	rad

Note that the bound on the departure ΔV is quite large and include a very strong minima at around 1- 4 km/sec (1:1 Earth orbit resonance) that often tricks the optimisers. Clearly, by reducing this bound (knowledge-based pruning) one can drastically help any optimiser to locate the correct global optima. As here we are interested in the algorithmic perfor-

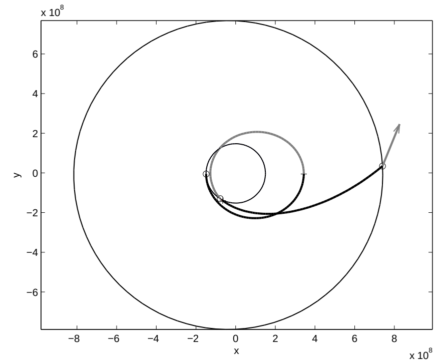

Figure 3: Best solution known for the problem SAGAS.

mances, we selected the bounds to create an interesting optimisation problem rather than to simplify the problem as much as possible and thus we included zones of the search space one could prune out by experience. The best solution known to this problem is $\mathbf{x}$ = [7020.49, 5.34817, 1, 0.498915, 788.763, 484.349, 0.4873, 0.01, 1.05, 10.8516, -1.57191, -0.685429], corresponding to a final objective function value of 18.1923 years, see Figure 3.

Cassini2 (EdVdVdEdJdS.mat)

In the section named Cassini1 we presented a global optimisation problem related to the mission Cassini. In that problem the description of the trajectory was made . In this section we consider a different model for the Cassini trajectory. This time we will allow for deep space maneuvers between each one of the planets and thus and MGADSM problem. This leads to a higher dimensional problem with a much higher complexity [5, 6]. We also consider, in the objective function evaluation, a rendezvous problem rather than an orbital insertion as in Cassini1. This is the main cause for the higher objective function values reached. For the twelve dimension state vector we use the bounds given in Table 4. These are consistent with the ones used in [4].

No other constraints are considered for this problem. The best known solution is $\mathbf{x}$ = [-815.144, 3, 0.623166, 0.444834, 197.334, 425.171, 56.8856, 578.523, 2067.98, 0.01, 0.470415, 0.01, 0.0892135, 0.9, 1.05044, 1.38089, 1.18824, 76.5066, $-1.57225, -2.01799, -1.52153, -1.5169$] corresponding to an objective function value of 8.92401 km/sec, see Figure 4. In the paper by Vasile and De Pascale [4] an objective function of 9.016 is reached for a similar problem, which is very close to the best solution we found. However, details in the implementation of the problem can easily account for quite significant variation of the objective value reached (for example planetary ephemerides, different bounds on some of the variables, etc.).

Table 4: State vector bounds in Cassini2

State	Variable	LB	UB	Units
x(1)	t_0	-1000	0	MJD2000
x(2)	V_∞	3	5	km/sec
x(3)	u	0	1	n/a
x(4)	v	0	1	n/a
x(5)	T_1	100	400	days
x(6)	T_2	100	500	days
x(7)	T_3	30	300	days
x(8)	T_4	400	1600	days
x(9)	T_5	800	2200	days
x(10)	η_1	0.01	0.9	n/a
x(11)	η_2	0.01	0.9	n/a
x(12)	η_3	0.01	0.9	n/a
x(13)	η_4	0.01	0.9	n/a
x(14)	η_5	0.01	0.9	n/a
x(15)	$\bar{r}_{p1}$	1.05	6	n/a
x(16)	$\bar{r}_{p2}$	1.05	6	n/a
x(17)	$\bar{r}_{p3}$	1.15	6.5	n/a
x(18)	$\bar{r}_{p4}$	1.7	291	n/a
x(19)	b_{incl1}	$-\pi$	π	rad
x(20)	b_{incl2}	$-\pi$	π	rad
x(21)	b_{incl3}	$-\pi$	π	rad
x(22)	b_{incl4}	$-\pi$	π	rad

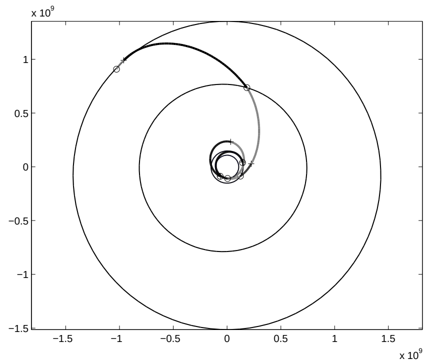

Figure 4: Best solution known for the problem Cassini2.

Messenger (EdEdVdVdMe.mat)

This trajectory optimisation problem represents a rendezvous mission to Mercury modelled as an

Table 5: State vector bounds in Messenger

State	Variable	LB	UB	Units
x(1)	t_0	1000	4000	MJD2000
x(2)	V_∞	1	5	km/sec
x(3)	u	0	1	n/a
x(4)	v	0	1	n/a
x(5)	T_1	200	400	days
x(6)	T_2	30	400	days
x(7)	T_3	30	400	days
x(8)	T_4	30	400	days
x(9)	η_1	0.01	0.99	days
x(10)	η_2	0.01	0.99	n/a
x(11)	η_3	0.01	0.99	n/a
x(12)	η_4	0.01	0.99	n/a
x(13)	$\bar{r}_{p1}$	1.1	6	n/a
x(14)	$\bar{r}_{p2}$	1.1	6	n/a
x(15)	$\bar{r}_{p3}$	1.1	6	n/a
x(16)	b_{incl1}	$-\pi$	π	n/a
x(17)	b_{incl2}	$-\pi$	π	n/a
x(18)	b_{incl3}	$-\pi$	π	n/a

MGADSM problem. The selected fly-by sequence is the same used in the first part of the Messenger mission. It is well known that a significant reduction of the required ΔV is possible if a number of resonant fly-bys follow the first Mercury encounter. Here we did not include that part of the trajectory in the optimisation problem as the dynamical model needed to represent multiple revolution solutions was not present in the code we planned to put on-line. We plan to publish the full trajectory problem description in a future work. For the eighteen dimensional global optimisation problem we consider the bounds listed in Table 5.

The best solution known for this problem is $\mathbf{x}$ = [2363.36, 1.68003, 0.381885, 0.512516, 400, 173.848, 224.702, 211.803, 0.238464, 0.265663, 0.149817, 0.485908, 1.34411, 3.49751, 1.1, 1.29892, 2.49324, 1.81426] with the objective value 8.981973 km/s. This result is shown in Figure 5.

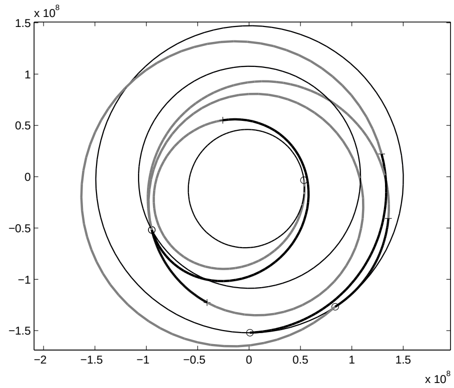

Figure 5: Best solution known for the problem Messenger.

Rosetta (EdEdMdEdEdA.mat)

The problem presented in this section is a MGADSM problem relative to a mission to the comet 67P/Churyumov-Gerasimenko. The fly-by sequence selected is similar to the one planned for the spacecraft Rosetta. The objective function considered is the total mission ΔV, including the launcher capabilities. The bounds used for the twenty-two dimension decision vector are listed in table 6. These are consistent with the ones used in [4].

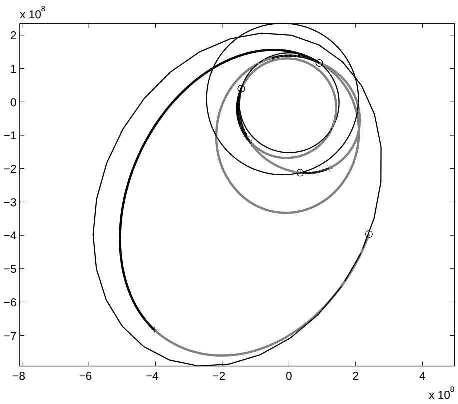

Figure 6: Best solution known for the problem Rosetta.

Table 6: State vector bounds in problem Rosetta

State	Variable	LB	UB	Units
x(1)	t_0	1460	1825	MJD2000
x(2)	V_∞	3	5	km/sec
x(3)	u	0	1	n/a
x(4)	v	0	1	n/a
x(5)	T_1	300	500	days
x(6)	T_2	150	800	days
x(7)	T_3	150	800	days
x(8)	T_4	300	800	days
x(9)	T_5	700	1850	days
x(10)	η_1	0.01	0.9	n/a
x(11)	η_2	0.01	0.9	n/a
x(12)	η_3	0.01	0.9	n/a
x(13)	η_4	0.01	0.9	n/a
x(14)	η_5	0.01	0.9	n/a
x(15)	$\bar{r}_{p1}$	1.05	9	n/a
x(16)	$\bar{r}_{p2}$	1.05	9	n/a
x(17)	$\bar{r}_{p3}$	1.05	9	n/a
x(18)	$\bar{r}_{p4}$	1.05	9	n/a
x(19)	b_{incl1}	$-\pi$	π	rad
x(20)	b_{incl2}	$-\pi$	π	rad
x(21)	b_{incl3}	$-\pi$	π	rad
x(22)	b_{incl4}	$-\pi$	π	rad

The best solution found for this problem is $\mathbf{x} = [1524.25, 3.95107, 0.738307, 0.298318, 365.123, 728.902, 256.049, 730.485, 1850, 0.199885, 0.883382, 0.194587, 0.0645205, 0.493077, 1.05, 1.05, 1.05, 1.36925, -1.74441, 1.85201, -2.61644, -1.53468]$, which correspond to an objective function value of $J(\mathbf{x}) = 1.4491$ km/s, see Figure 6.

The Optimisers

We here try some widely used global optimisation algorithms on the benchmark problems described above. In this section a short introduction is given on the solvers used in the paper.

Differential Evolution (DE) This optimisation algorithm is based on updating each element of a set (population) of feasible solutions by using the weighted difference of two (or more) other randomly selected population elements. This way no separate probability distribution has to be used which makes the scheme completely self-organizing [11].

Particle Swarm Optimization (PSO) This is another population-based algorithm inspired by the social behaviour of bird or fish flockings [12]. In a PSO method, each element (particle) evolves by taking the combination of the current global best and individual best solutions into account.

Genetic Algorithm (GA) The goodness of an individual in the population is measured by its fitness value (i.e. the objective function value). GA evaluates the fitness of each individual in the population and then while not converged it selects individuals to reproduce, performs crossover and mutation to make the offspring evaluates the individual fitnesses of the offspring and finally replaces the worst ranked part of the population with the offspring [13].

Adaptive Simulated Annealing (ASA) Simulated Annealing (SA) [14] picks some neighbour y of a point x and compute its energy (this is like the fitness value in the above algorithms). SA moves to this new point y based on a randomly selected number which depends on the distance of the corresponding function values and on a global parameter T (temperature), that is gradually decreased during the process. Adaptive simulated annealing [15] is a variant of SA in which the algorithm parameters that control temperature schedule and random step selection are automatically adjusted according to algorithm progress.

GLOBAL This multistart clustering global optimization method [16] is based on Boender's algorithm [17]. It's goal is to find all local minimizer points that are potentially global. These local minimizers will be found by means of a local search procedure, starting from appropriately chosen points from the sample drawn uniformly within the set of feasibility. In an effort to identify the region of attraction of a local minimum, the procedure invokes a clustering procedure. In this paper we use the recent version of this solver implemented in MATLAB and documented in [18].

COOP Recent results show that collaborative usage of population based algorithms can lead to performance improvement [7]. This algorithmic scheme is a combination of DE and PSO in a cooperative way. Namely, we allow N_1 iterations for DE and the result population is passed to the PSO as initial value. Then PSO can take N_2 iterations and the result passed back to DE which does again N_1 iterations and passes the result back to PSO, and so on. Typically $N_1 = N_2$ and they are much smaller than the maximum iterations allowed.

The Results

The results obtained with the solvers listed in the previous section is reported here. For all the solvers 20 independent run was made and the best and the average objective values are documented. The allowed number of objective function evaluations (NFE) were limited in the following way:

Model	NFE
Cassini1	20,000
GTOC1	25,000
SAGAS	20,000
Cassini2	30,000
Messenger	30,000
Rosetta	35,000

Note that this is an ad-hoc set up, using more function evaluations should lead to better results for some of the algorithms. However, our goal was to make a fair comparison of the solvers and for this purpose a meaningful stopping criteria was needed to be fixed.

In case of GLOBAL there are some built-in stopping criteria which would be fulfilled earlier than the limit in the number of function evaluations is reached. This feature was kept in this benchmarking, so occasionally GLOBAL stopped the run before it reached the maximum NFE limit.

The following option were set for the different solvers:

- DE: population size: 20, $F = 0.95$, $CR = 0.9$, strategy: rand/1/bin.
- PSO: population size: 20, $\omega = 0.65$, $\eta_1 = \eta_2 = 2$, $v_{\max} = 0.5$.
- GA: population size: 20, crossover rate: 0.75, mutation rate: 0.05.
- ASA: default options were used.
- GLOBAL: $N100 = 5,000$, $NG0 = 10$, $NSIG = 8$, local minimizer: UNIRANDI.
- COOP: $N_1 = N_2 = 50$ and the same parameters were used as the stand alone versions of DE and PSO.

These values are more or less the standard ones usually used for the corresponding solvers. Note that changing these parameters would also lead different results. Here we do not address the problem of finding a good (or optimal) parameter setup for a particular method for the trajectory optimisation problems.

MGA problems

First, the results for the MGA problems (Cassini1 and GTOC1) are reported in Table 7. None of the solvers was able to find the reported best solutions.

In case of Cassini1 the best results were given by DE and COOP, although they stacked at 5.303 which is a very strong local optimum point of this particular problem. In terms of average performance GLOBAL was the best for this problem.

For the GTOC1 problem (note that this is a maximisation problem) the ASA and GLOBAL performed relatively well compared to the other solvers. However, as it was expected, COOP reached much higher objective value both as best and as average. This result is consistent to the one obtained in [7] where a quite similar problem was considered and a cooperative strategy was tested.

Table 7: Results obtained for the MGA problems. For all the solvers the best and the average (out of 20 independent run) objective values are reported.

Solver	Cassini1		GTOC1	
	best	average	best	average
DE	5.3034	8.8036	891,757	342,613
PSO	6.1857	11.7060	582,169	295,468
GA	5.5973	13.1635	871,281	321,119
ASA	5.7806	15.5844	1,173,444	547,372
GLOBAL	5.5243	6.6703	1,035,278	609,649
COOP	5.3034	11.3677	1,365,286	790,252

MGADSM problems

For these more difficult problems the benchmarking results are reported in Table 8 and 9. Again, none of the solvers were able to find the best solutions known. Maybe one can obtain better results allowing more function evaluations, but it seems that a tendency can be seen even in this case.

The SAGAS problem seems to be the most challenging one to the solvers. As best solution among the tested methods ASA gave the lowest function value and that was the best algorithm in average. These results compared to the best solution known (18.1923) leads to the consequence that all the solvers have difficulties to find an acceptable solution to this problem.

For the DSM version of the Cassini mission DE gives the best solution, which is significantly better then the results of the stand alone other solvers. However, in terms of average performance ASA and GLOBAL are better. COOP reached almost as good best value as DE and in average it is the best solver for this problem.

Considering the Messenger mission DE gave again the best result, while ASA and GLOBAL did again good performance in average. As for the previous problems, COOP was the best solver in average.

For the Rosetta mission DE was able to find a very low function value. In this case DE performed quite well in average, followed by ASA. COOP was again, in average, this the best among the other solvers.

Table 8: Results obtained for the MGADSM problems SAGAS and Cassini2. For all the solvers the best and the average (out of 20 independent run) objective values are reported.

Solver	SAGAS		Cassini2	
	best	average	best	average
DE	712.940	1234.60	14.9567	27.0123
PSO	1131.746	1555.92	27.6783	31.1449
GA	1283.45	1657.38	27.4809	32.4357
ASA	29.621	741.84	19.8575	24.3598
GLOBAL	416.63	1072.57	22.0915	25.2963
COOP	846.219	1033.01	15.7216	22.7855

Table 9: Results obtained for the MGADSM problems Messenger and Rosetta. For all the solvers the best and the average (out of 20 independent run) objective values are reported.

Solver	Messenger		Rosetta	
	best	average	best	average
DE	11.5660	20.5040	2.2191	8.5766
PSO	17.8808	19.9091	12.5653	16.3658
GA	18.9884	21.7286	15.6275	18.0396
ASA	14.6387	17.3892	6.3417	12.8708
GLOBAL	17.1259	18.8728	8.2186	11.8180
COOP	13.0425	15.4839	2.9308	8.3039

Conclusions

We introduce six global trajectory optimisation problems in the hope they may serve as reference problems to further development of models and solvers. We give preliminary test results for these problems obtained with standard global optimisation solvers. We find that in almost every case the simple application of standard global optimisation solvers is not enough to find good solutions and some more elaborated approach is desirable. The simple scheme proposed in this paper to use collaboratively different solvers (differential evolution and particle swarm optimisation were used in here), improves the average performance of the stand alone solvers and is well suited to find good solutions to trajectory optimisation problems. A similar approach, only distributed and including more solvers, was used to present a 'best' solution for each one of the proposed problems.

References

[1] Petropoulos, A., Longuski, J., and Bonfiglio, E., *Journal of Spacecraft and Rockets*, Vol. 37, 2000, pp. 776–783.

[2] Hartmann, J., Coverstone-Carroll, V., and Williams, S., "Generation of Optimal Spacecraft Trajectories via a Pareto Genetic Algorithm," *Journal of the Astronautical Sciences*, Vol. 46, 1998, pp. 267–282.

[3] Izzo, D., Becerra, V., Myatt, D., Nasuto, S., and Bishop, J., "Search Space Pruning and Global Optimisation of Multiple Gravity Assist Spacecraft Trajectories," *Journal of Global Optimisation*, Vol. 38, 2007, pp. 283–296.

[4] Vasile, M. and De Pascale, P., "Preliminary Design of Multiple Gravity-Assist Trajectories," *Journal of Spacecraft and Rockets*, Vol. 42, 2006, pp. 794–805.

[5] Myatt, D. R., Becerra, V. M., Nasuto, S. J., and Bishop, J. M., "Advanced Global Optimisation Tools for Mission Analysis and Design," Ariadna Final Report 03-4101a, European Space Agency, the Advanced Concepts Team, 2004, Available on line at www.esa.int/act.

[6] Di Lizia, P. and Radice, G., "Advanced Global Optimisation Tools for Mission Analysis and Design," Ariadna Final Report 03-4101b, European Space Agency, the Advanced Concepts Team, 2004, Available on line at www.esa.int/act.

[7] Vinkó, T., Izzo, D., and Pinna, F., "Learning the Best Combination of Solvers in a Distributed Global Optimisation Environment," *Submitted for publication*, 2007.

[8] Izzo, D., "Advances in Global optimisation For Space Trajectory Design," Paper ISTS 2006-d-45, 25th International Symposium on Space Technology and Science, Japan, 2006.

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

[10] Izzo, D., "1st ACT global trajectory optimisation competition: Problem description and summary of the results," *Acta Astronautica*, Vol. 61, 2007, pp. 731–734.

[11] Storn, R. and Price, K., "Differential Evolution - a Simple and Efficient Heuristic for Global Optimizationover Continuous Spaces," *Journal of Global Optimization*, Vol. 11, 1997, pp. 341–359.

[12] Kennedy, J. and Eberhart, R. C., "Particle swarm optimization," *Proceedings of IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.

[13] Holland, J. H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.

[14] Kirkpatrick, S., Gelatt, C., and Vecchi, M., "Optimization by simulated annealing," *Science*, Vol. 220, 1983, pp. 671–680.

[15] Ingber, L., "Very fast simulated re-annealing," *Mathematical and Computer Modelling*, Vol. 12, 1989, pp. 967–973.

[16] Csendes, T., "Nonlinear parameter estimation by global optimization – efficiency and reliability," *Acta Cybernetica*, Vol. 8, 1988, pp. 361–370.

[17] Boender, C., Rinnooy Kan, A., Timmer, G., and Stougie, L., "A stochastic method for global optimization," *Mathematical Programming*, Vol. 22, 1982, pp. 125–140.

[18] Pál, L. and Csendes, T., "Improvements on the GLOBAL Optimization Algorithm with Numerical Tests," *Submitted for publication*, 2007.