MIDACO Software Performance on Interplanetary Trajectory Benchmarks

Martin Schlueter*

Institute of Space and Astronautical Science,
Japan Aerospace Exploration Agency (JAXA),
3-1-1 Yoshinodai, Sagamihara, Kanagawa 252-5210, Japan

Abstract

A numerical study of the MIDACO optimization software on the well known GTOP benchmark set, published by the European Space Agency (ESA), is presented. The GTOP database provides trajectory models of real-world interplanetary space missions such as Cassini, Messenger or Rosetta. The trajectory models are formulated as constrained nonlinear optimization problems and are known to be difficult to solve.

Here a comprehensive and rigorous numerical analysis of the MIDACO out-of-the-box performance on the GTOP benchmark set is presented and discussed. In the past, the putative best known solutions of these benchmarks often required several months and even years to be found. In this contribution it will be shown, that MIDACO is able to solve five out of seven of these benchmarks to their best known solution within minutes to hours.

Keywords: Interplanetary Space Trajectory; Optimization; GTOP; Cassini; Messenger; Rosetta; MIDACO; Parallelization

*Corresponding author
Email address: `martin@flab.isas.jaxa.jp` (Martin Schlueter)

1. Introduction

Since 2005 the Advanced Concept Team (ACT) of the European Space Agency (ESA) publishes a database of Global Trajectory Optimization (GTOP) problems (see ESA Website (2013)) and encourages researchers from space engineering and operational research to submit putative best known solutions to these benchmarks. This benchmark set is known as the GTOP database and represents trajectory models of real-world interplanetary space missions. Currently the GTOP database consists of eight benchmark problems, which are published as a single problem instance except the *Tandem* benchmark, which is published in 50 different instances. In this contribution the entire GTOP set, except the Tandem problem, is considered. The *Tandem* problem is excluded, because its variation of 50 individual instances is unsuitable for the proposed numerical test setup, which considers 10 test runs of up to 24 hours for each problem instance[1]. The considered seven GTOP benchmark problems are listed in Table 1 together with the number of decision variables, the number of constraints, the number of submissions of putative best known solutions and the time between the first and last of such solution submission.

Table 1: GTOP database benchmark problems

Benchmark Name	Variables	Constraints	Number of submissions	Time between first and last submission
Cassini1	6	4	3	6 month
GTOC1*	8	6	2	13 month
Messenger (reduced)	18	0	3	11 month
Messenger (full)*	26	0	9	58 month
Cassini2*	22	0	7	14 month
Rosetta	22	0	7	6 month
Sagas	12	2	1	-

*Best known solution found by MIDACO and published on ESA Website (2013).

The GTOP benchmark problems are all highly non-linear and are known to be difficult to solve, despite their relatively low number of decision variables and constraints. The difficulty of these problems is also reflected in the

[1]In case of the Tandem benchmark this setup would imply a calculation time of up to 500 days.

time span between the first and last solution submission reported in Table 1. The easiest problem in the set is considered to be Cassini1, while the most difficult problem in the set is considered to be Messenger (full), which took nearly five years between its first and last solution submission.

Note that the GTOP benchmarks are in fact so difficult, that the majority of currently available publications focus only on a few benchmarks (often just one or two instances). Table 2 gives an overview on some publications that consider GTOP benchmark problems. In Table 2 it is stated how many problems were considered by the authors and how many of those could be solved to the current best known solution[1].

Table 2: Overview on publications on GTOP database

Author(s)	Problems	Solved
Gruber (2009)	1	1
Lancinskas, Zilinskas & Ortigosa (2010)	1	0
Danoy, Pinto & Dorronsoro (2012)	1	1
Islam, Roy & Suganthan (2012)	2	0
Gad (2011)	2	0
Ampatzis and Izzo (2009)	2	1
Biazzini, Banhelyi, Montresor et al. (2009)	2	1
Musegaas (2012)	2	2
Henderson (2013)	2	1
Biscani, Izzo & Yam (2010)	3	2
Izzo (2010)	4	1
Addis, Cassioli, Locatelli et al. (2011)	4	3
Vinko & Izzo (2008)	5	1
Stracquadanio, La Ferla, De Felice et al. (2011)	7	6

From Table 2 it can be seen, that only Stracquadanio, La Ferla, De Felice et al. (2011) considered (nearly) the full GTOP testbed. Despite impressive final objective function values presented in that reference it is to note, that no information on the required time, evaluation or possible tuning of the algorithm and problem search space is given in that reference.

[1] It is referred here to the best known solution due to the date of this publication within a precision of 0.1%. Therefore some of the solutions presented in the references of Table 2 as best known are nowadays known to be sub-optimal.

The MIDACO software was developed as a general purpose optimization solver and has been written entirely from scratch by the author. The software does not rely on any external libraries. It has been extensively tested on space applications and in particular trajectory optimization (see for example Schlueter (2012b) or Schlueter, Erb, Gerdts et al. (2013a)). Currently MIDACO holds the best known solution to three of the benchmarks listed in Table 1, including the best known solution to the most difficult problem: Messenger (full). MIDACO is based on the concept of evolutionary programming, which aims on approximating a good solution to difficult problems in a reasonable time. MIDACO employs the Ant Colony Optimization (ACO) metaheuristic in combination with the Oracle Penalty Method. Readers who are interested in the detailed theoretical ACO algorithm used in MIDACO, are referred to Schlueter, Egea & Banga (2009) and Schlueter (2012b), Chapter 2 and in particular Section 2.4, where an illustrative step-by-step example is given. Readers who are interested in the details of the Oracle Penalty Method are referred to Schlueter & Gerdts (2010). Note that the performance strength of MIDACO does not only rely on its fundamental algorithms, but also on its sophisticated software implementation which took over 7 years of development for the current version.

The purpose of this contribution is to rigorously demonstrate the ***out-of-the-box***[1] performance capabilities of the MIDACO software on space mission trajectory optimization problems and to provide a performance reference for other researchers working on the GTOP database. This paper is structured as follows: In Section 2, numerical results on all seven benchmarks are presented and discussed in detail in individual subsections. In Section 3, a brief summary of the numerical results obtained from Section 2 is displayed as an overview. The paper finishes with some conclusions and an outlook on future research.

[1]The term out-of-the-box means here, that the software is used with all its default settings and no effort has been undertaken to specifically tune any of its parameters. Furthermore the benchmarks are considered "as is" and are not modified in any way.

2. Numerical Results

This section presents the numerical results obtained by MIDACO on the GTOP database. In regard to an ***out-of-the-box*** approach, every individual benchmark is tested under exact identical conditions. In order to avoid unnecessary repeating of those test setup details, it should be given in this introduction in full detail. The testing for each benchmark was split into two setups: A main-setup and a refinement-setup.

The main-setup consists of 10 individual MIDACO tests runs on the benchmark problem. The main-setup does not include any algorithmic parameter tuning for MIDACO and therefore represents the out-of-the-box performance. Each test run of the main-setup is either stopped, when MIDACO reaches a solution which is at least 0.1%[1] close to the best known one, or if a maximal cpu-time budget of 86,400 seconds (24 hours) is reached. Each test run of the main-setup uses a different random seed, starting from 0 to 9 for the ten test runs. This means, no particular good random seeds were picked.

The refinement-setup consist of only a single test run of MIDACO, using the best solution found in the previous main-setup as starting point. As the purpose of the refinement-setup is to improve the precision of the previously gained solution, one specific algorithmic parameter of MIDACO is tuned. This is the FOCUS parameter (see the MIDACO user manual Schlueter & Munetomo (2013c)), which disables global search heuristics within MIDACO and allows the algorithm to focus its search effort on the current best solution. The value for the FOCUS parameter was fixed to -10000 for all benchmarks. Note that this parameter was therefore <u>not</u> individually tuned for each benchmark. Every refinement test run was started with the default random seed (zero).

All benchmarks are considered in their original form provided on the ESA Website (2013). This means in particular, that no modification of the search space (the lower and upper bounds for decision variables) has been conducted. As starting point for the test runs of the main-setup, the lower bounds were used in all cases. For constrained problems, MIDACO's default tolerance for constraint violation was considered, which is 10^{-3}.

[1]Note that 0.1% is the official precision requested by ESA for new solution submissions. Solutions close as 0.1% can therefore be considered "as good as" the solutions published on the ESA Website (2013).

The C/C++ version of MIDACO 4.0 was used for the numerical tests and the parallelization feature of MIDACO based on openMP was enabled. The parallelization feature of MIDACO aims on executing several solution iterates in so called "blocks". As the tests were performed on a quad-core cpu with hyper-threading, a parallelization factor of 8 was applied. This implies that each block corresponds to 8 individual function evaluations. In the following tables both, the number of blocks and the number of function evaluation are reported. Because the GTOP benchmarks are fairly cheap to compute regarding their cpu-time, the speed-up factor of this parallelization approach is not fully effective, but is sufficient to give a speed up between two and three times in contrast to a serial execution. Details on the parallelization concept of MIDACO can be found in Schlueter, Gerdts & Rueckmann (2012a) or Schlueter & Munetomo (2013b).

All presented numerical results were calculated on the same PC with an Intel Core i7 quad-core CPU 920@2.67 GHz clockrate and 4GB RAM memory running Linux (CentOS 6.3). In regard to full transparency and reproducibility, the main file source code, the makefile, the final solutions and the numerical screen outputs of all test runs are made publicly available at:

`http://www.midaco-solver.com/index.php/about/benchmarks/esa-gtop`

Note that the total cpu-time required for all tests presented here took over 41 days.

2.1. Results on Cassini1

The Cassini1 benchmark models an interplanetary space mission to Saturn. The objective of the mission is to get captured by Saturn's gravity into an orbit having a pericenter radius of 108,950 km and an eccentricity of 0.98. The sequence of fly-by planets for this mission is given by Earth-Venus-Venus-Earth-Jupiter-Saturn, whereas the first item is the start planet and the last item is the final target. This benchmark does not include deep space maneuvers (DSM) and is therefore easier than the Cassini2 benchmark. The objective function of this benchmark is to minimize the total deltaV accumulated during the mission, including the launch and capture manouvre. The benchmark invokes 6 decision variables, which are described as follows:

This benchmark further considers four constraints, which impose an upper limit on the pericenters for the four fly-by manourvers. The currently best

Variable	Description
1	Initial day measured from 1-Jan 2000
$2 \sim 6$	Time interval between events (e.g. departure, fly-by, capture)

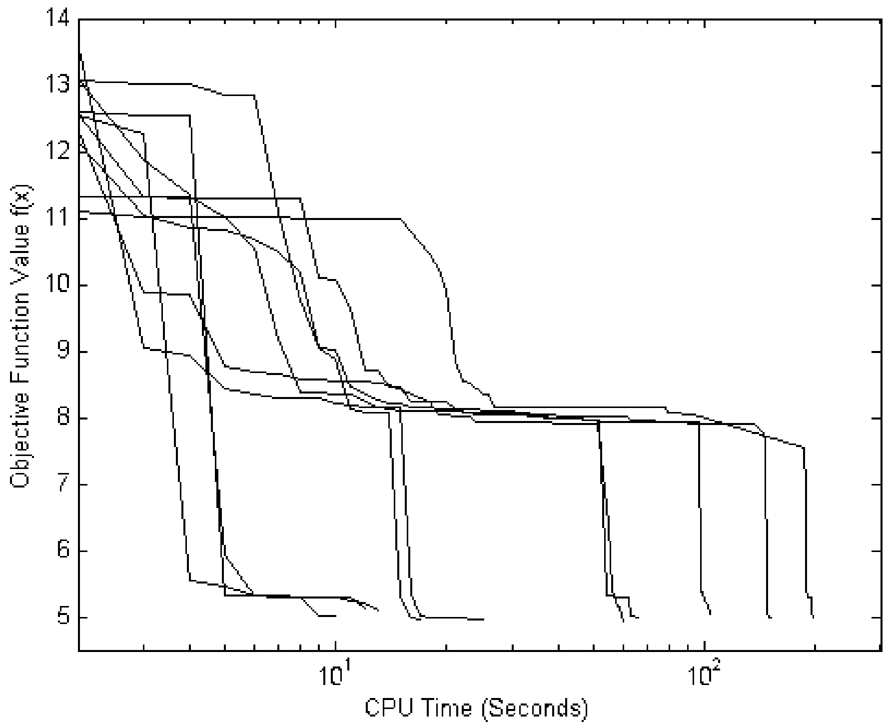
known solution published on the ESA Website (2013) was provided by a particle swarm optimization algorithm in 2006 and corresponds to an objective function value of about $f(x) = 4.93073$. This benchmark is considered the easiest one in the GTOP database. Table 3 displays the numerical results of 10 individual test runs of MIDACO on Cassini1.

Table 3: 10 Test runs of MIDACO on Cassini1

Run	$f(X)$	*Blocks*	*Eval*	*Time (Sec)*
1	4.933760	226,556	1,812,448	12
2	4.935615	1,290,295	10,322,360	66
3	4.935378	189,257	1,514,056	10
4	4.935624	328,743	2,629,944	17
5	4.935491	1,050,408	8,403,264	60
6	4.935602	2,019,317	16,154,536	103
7	4.934514	2,926,304	23,410,432	152
8	4.935575	240,084	1,920,672	13
9	4.935630	471,085	3,768,680	25
10	4.934698	865,179	6,921,432	198

The convergence curves of the 10 individual runs from Table 3 are illustrated in Figure 1. Note that the X-axis in Figure 1 is given in logarithmic scale.

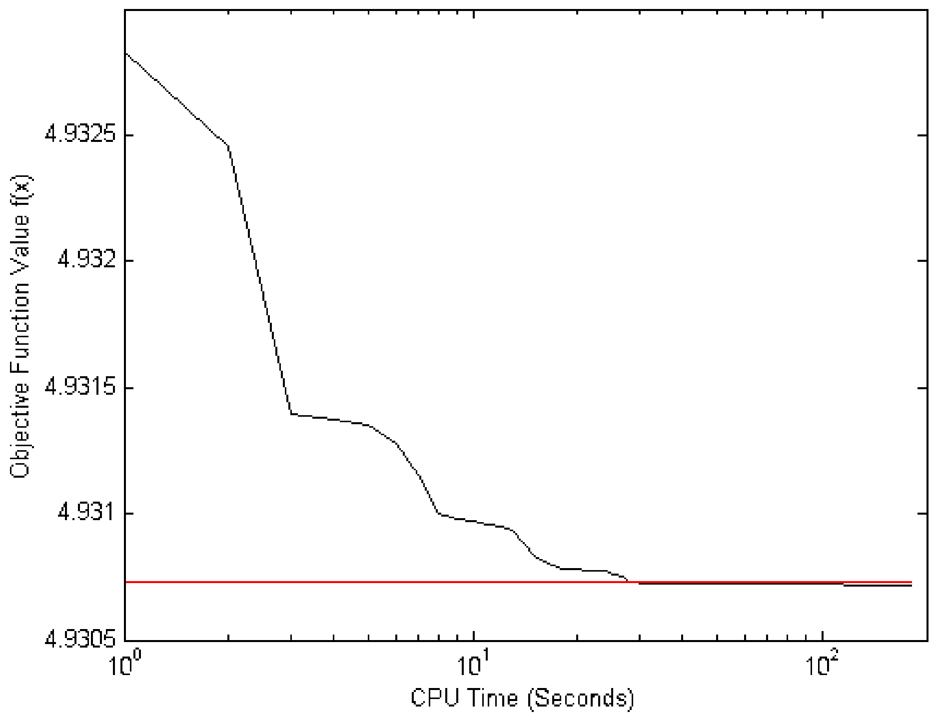
Figure 1: Convergence curves of 10 runs by MIDACO on Cassini1



From the results in Table 3 it can be seen that MIDACO is able to reach the best known solution within a precision of 0.1% in all cases. Test run number 3 represents the best run corresponding to a cpu-time of 10 seconds and 189,257 processed *Blocks*. Test run number 10 represents the worst run corresponding to a cpu-time of 198 seconds and 865179 processed *Blocks*. From Figure 1 it can be seen that after around 20 seconds all test runs converged to a objective function value less than $f(x) = 9.89$ (which is around twice as high as the best known solution).

Figure 2 illustrates the convergence curve of the refinement run for the solution from test run number 3 of Table 3. In addition to the convergence curve, the value of the best known solution is plotted as a red line in Figure 2. Note that the X-axis in Figure 2 is given in logarithmic scale.

Figure 2: Convergence curves of refinement run on best solution from Table 3



From Figure 2 it can be seen that the exact value of the best known solution is (b)reached at around 30 seconds. Furthermore it can be seen from Figure 2 that no further improvements are achieved after around 100 seconds. The refined MIDACO solution represents a (very marginal) improvement of about 0.0004% in respect to the current best known solution. The constraint violation of the solution is zero.

For further numerical results of MIDACO on Cassini1, see Schlueter & Munetomo (2014).

2.2. Results on GTOC1

The GTOC1 benchmark models a multi gravity assist space mission to asteroid TW229. The mission model drew inspiration from the first edition of the Global Trajectory Optimisation Competition (GTOC) held by ESA in 2007, see Izzo (2007). The objective of the mission is to maximize the change in the semi-major axis of the asteroid orbit.The sequence of fly-by planets for this mission is given by Earth-Venus-Earth-Venus-Earth-Jupiter-Saturn-TW229, whereas the first item is the start planet and the last item is the final target. This benchmark invokes 8 decision variables which are described as follows:

Variable	Description
1	Initial day measured from 1-Jan 2000
$2 \sim 8$	Time interval between events (e.g. departure, fly-by, capture)

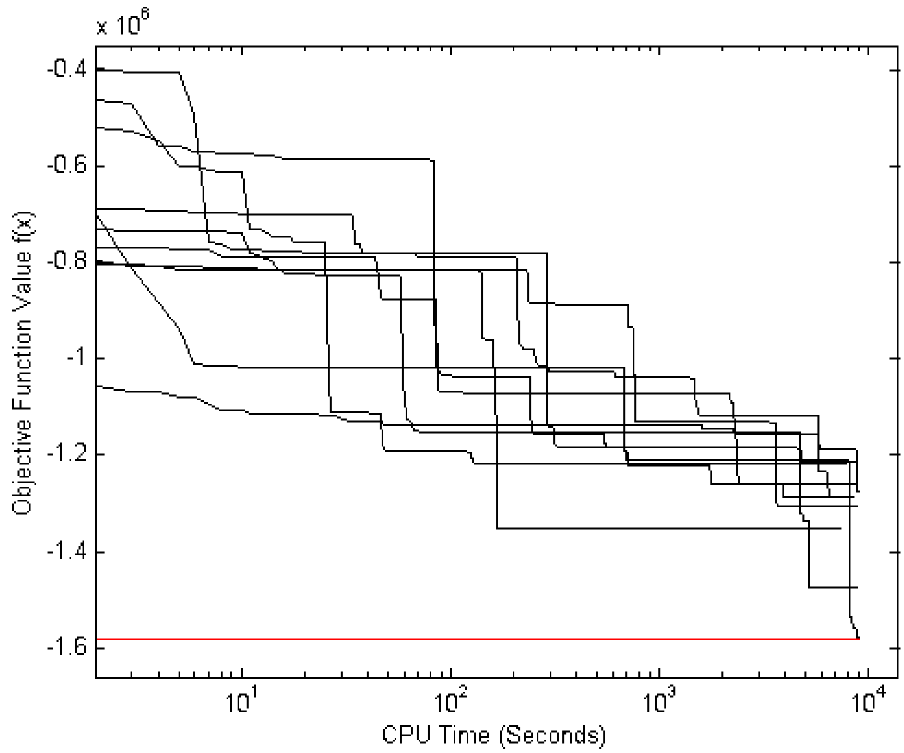
This benchmark further considers four constraints, which impose an upper limit on the pericenters for the four fly-by manourvers. The currently best known solution published on the ESA Website (2013) was provided by MIDACO in 2010 and corresponds to an objective function value of about $f(x) = -1,581,950$. Table 4 displays the numerical results of 10 individual test runs of MIDACO on GTOC1. In contrast to all other considered benchmark problems, the cpu-time budget for GTOC1 was reduced from 24 hours to 2.5 hours, due to numerical problems that appeared during the test runs running for longer than some hours. Those numerical problems caused the program to crash and despite some efforts, unfortunately this problem could not be debugged.

Table 4: 10 Test runs of MIDACO on GTOC1

Run	$f(x)$	*Blocks*	*Eval*	*Time (Sec)*
1	-1,576,138.804495	112,762,980	902,103,840	9,000
2	-1,473,037.748785	102,835,741	822,685,928	9,000
3	-1,352,425.329790	114,074,622	912,596,976	9,000
4	-1,216,521.106878	105,953,390	847,627,120	9,000
5	-1,287,827.674593	110,374,220	882,993,760	9,000
6	-1,212,631.180987	105,591,555	844,732,440	9,000
7	-1,274,590.172825	111,975,533	895,804,264	9,000
8	-1,304,656.895131	110,261,908	882,095,264	9,000
9	-1,287,827.597517	104,785,169	838,281,352	9,000
10	-1,258,275.058802	103,004,166	824,033,328	9,000

The convergence curves of the 10 individual runs from Table 4 are illustrated in Figure 3. In addition to the convergence curve, the value of the best known solution is plotted as a red line in Figure 3. Note that the X-axis in Figure 3 is given in logarithmic scale.

Figure 3: Convergence curves of 10 runs by MIDACO on GTOC1



From the results in Table 4 it can be seen that MIDACO is not able to reach the best known solution within a precision of 0.1% in any test run within the given time budget. Test run number 1 represents the best run corresponding to an objective function value of around -1,576,138. Test run number 6 represents the worst run corresponding to an objective function value of around -1,212,631. From both Table 4 and Figure 3 it can be seen that the cpu-time budget of 2.5 hours is too short for MIDACO to converge to a final solution. The numerical problems that forced us to reduce the cpu-time budget is therefore considered a misfortune, in particular as MIDACO has in the past provided the current best known solution. Because of the numerical problems, no refinement test run was performed for GTOC1.

2.3. Results on Messenger (reduced)

The Messenger (reduced) benchmark models an interplanetary space mission to Mercury and does not include resonant flyby's at Mercury. The sequence of fly-by planets for this mission is given by Earth-Earth-Venus-Venus-Mercury, whereas the first item is the start planet and the last item is the final target. The objective of this benchmark to be minimized is the total deltaV accumulated during the full mission. The benchmark invokes 18 decision variables which are described as follows (for details on hyperbolic trajectories, see Kemble (2006)):

Variable	Description
1	Initial day measured from 1-Jan 2000
2	Initial excess hyperbolic speed (km/sec)
3	Component of excess hyperbolic speed
4	Component of excess hyperbolic speed
$5 \sim 8$	Time interval between events (e.g. departure, fly-by, capture)
$9 \sim 12$	Fraction of the time interval after which DSM occurs
$13 \sim 15$	Radius of flyby (in planet radii)
$16 \sim 18$	Angle measured in planet B plane of the planet approach vector

The currently best known solution published on the ESA Website (2013) was provided by ESA itself in 2009 and corresponds to an objective function value of about $f(x) = 8.6305$. Table 5 displays the numerical results of 10 individual test runs of MIDACO on Messenger (reduced).

The convergence curves of the 10 individual runs from Table 5 are illustrated in Figure 4. Note that the X-axis in Figure 4 is given in logarithmic scale.

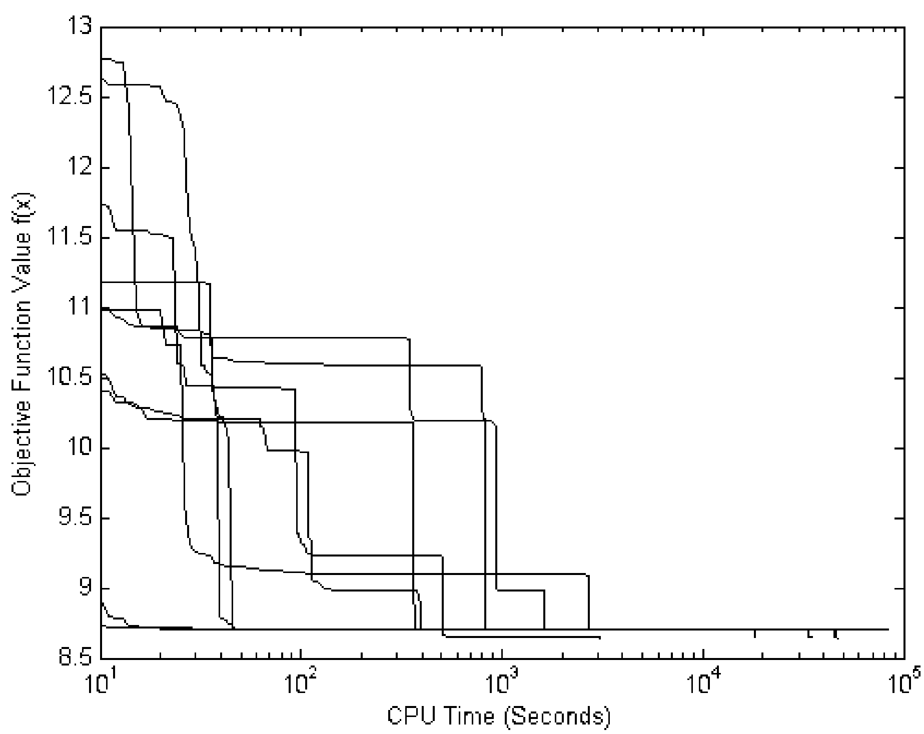
From the results in Table 5 it can be seen that MIDACO is able to reach the best known solution within a precision of 0.1% in 6 out of 10 cases. Test run number 7 represents the best run corresponding to a cpu-time of 3096 seconds and around 30 Million processed *Blocks*. In all 4 cases were MIDACO does not reach the best known solution within a precision of 0.1%, it obtains an objective function value of around $f(x) = 8.7016$, which appears to be a strong[1] local optimum. From Figure 4 it can be seen that

[1]By strong local optimum it is meant, that despite all restarts heuristics within MIDACO, the software was not able to escape from this local optimum. The attraction of such solution is therefore considered *strong*.

Table 5: 10 Test runs of MIDACO on Messenger (reduced)

Run	$f(x)$	$Blocks$	$Eval$	$Time\ (Sec)$
1	8.701630	849,630,359	6,797,042,872	86,400
2	8.638482	451,696,680	3,613,573,440	46,017
3	8.701631	846,510,501	6,772,084,008	86,400
4	8.701630	860,380,827	6,883,046,616	86,400
5	8.701630	849,812,316	6,798,498,528	86,400
6	8.638482	452,641,059	3,621,128,472	45,478
7	8.638620	30,732,566	245,860,528	3,096
8	8.638625	176,473,292	1,411,786,336	18,340
9	8.638571	358,532,292	2,868,258,336	35,304
10	8.638482	452,589,650	3,620,717,200	47,075

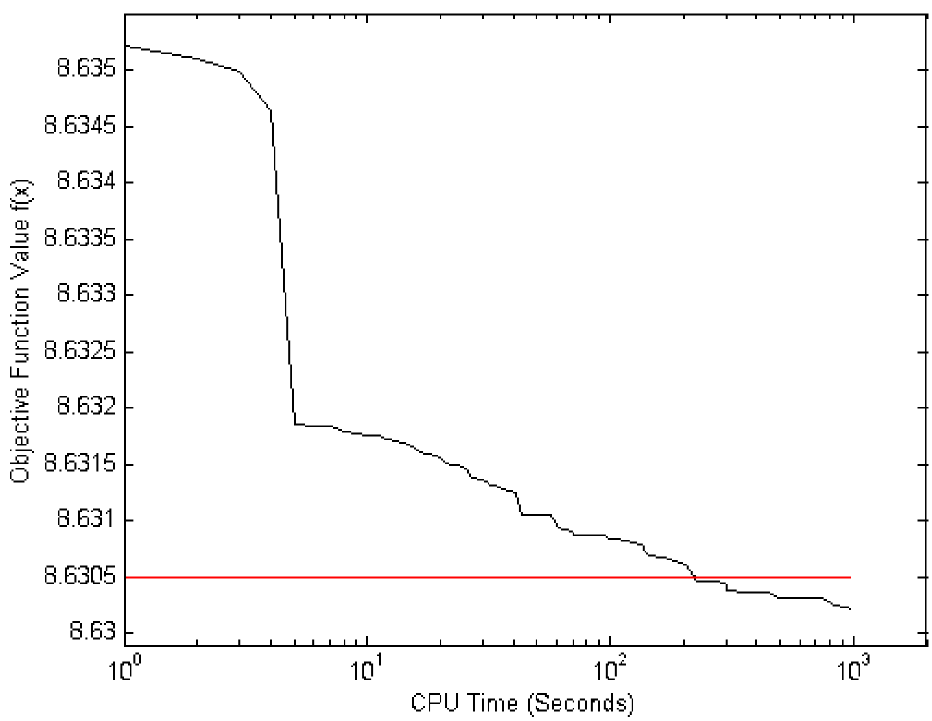
Figure 4: Convergence curves of 10 runs by MIDACO on Messenger (reduced)



after around 3000 seconds (less than an hour) all test runs converged to a objective function value which is less or around $f(x) = 8.7$ (which is closer than 1% on the best known solution).

Figure 5 illustrates the convergence curve of the refinement run for the solution from test run number 7 of Table 5. In addition to the convergence curve, the value of the best known solution is plotted as a red line in Figure 5. Note that the X-axis in Figure 5 is given in logarithmic scale.

Figure 5: Convergence curves of refinement run on best solution from Table 5



From Figure 5 it can be seen that the exact value of the best known solution is (b)reached at around 200 seconds. Furthermore it can be seen from Figure 5 that no further improvements are achieved after around 1000 seconds. The refined MIDACO solution represents a (very marginal) improvement of about 0.003% in respect to the current best known solution.

2.4. Results on Messenger (full)

The Messenger (full) benchmark models an interplanetary space mission to Mercury, including resonant flyby's at Mercury. The sequence of fly-by planets for this mission is given by Earth-Venus-Venus-Mercury-Mercury-Mercury-Mercury, whereas the first item is the start planet and the last item is the final target. The objective of this benchmark to be minimized is the total deltaV accumulated during the full mission. The benchmark invokes 26 decision variables which are described as follows (for details on hyperbolic trajectories, see Kemble (2006)):

Variable	Description
1	Initial day measured from 1-Jan 2000
2	Initial excess hyperbolic speed (km/sec)
3	Component of excess hyperbolic speed
4	Component of excess hyperbolic speed
$5 \sim 10$	Time interval between events (e.g. departure, fly-by, capture)
$11 \sim 16$	Fraction of the time interval after which DSM occurs
$17 \sim 21$	Radius of flyby (in planet radii)
$22 \sim 26$	Angle measured in planet B plane of the planet approach vector

The currently best known solution[2] published on the ESA Website (2013) was provided by MIDACO in April 2014 and corresponds to an objective function value of about $f(x) = 1.972$. This benchmark is considered currently the most challenging one in the GTOP database. Table 6 displays the numerical results of 10 individual test runs of MIDACO on Messenger (full).

The convergence curves of the 10 individual runs from Table 6 are illustrated in Figure 6. In addition to the convergence curve, the value of the best known solution is plotted as a red line in Figure 6. Note that the X-axis in Figure 6 is given in logarithmic scale.

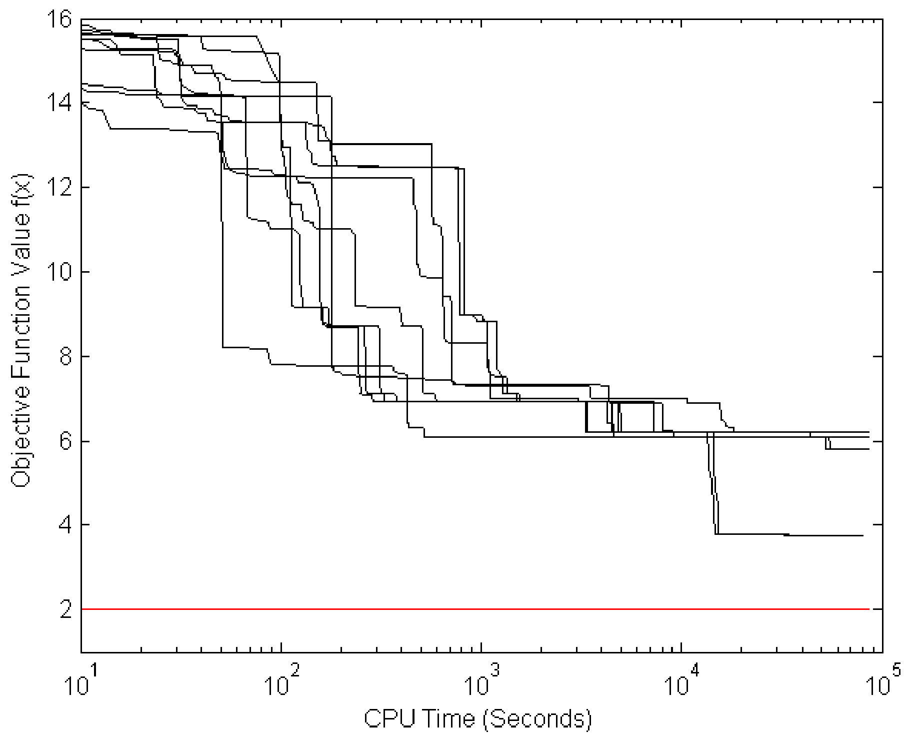
From the results in Table 6 it can be seen that MIDACO is not able to reach the best known solution within a precision of 0.1% in any test run. Test run number 6 represents the best run corresponding to an objective function value of around 3.774. From Figure 4 it can be seen that after around 2000 seconds (less than an hour) all test runs converged to a objective function

[2] Note that also the second and third best knwon solution to the Messenger (full) problem was provided by MIDACO. Finding these solutions required several weeks of calculation time.

Table 6: 10 Test runs of MIDACO on Messenger (full)

Run	$f(X)$	$Blocks$	$Eval$	$Time$ (Sec)
1	3.773516	259,926,212	2,079,409,696	86,400
2	5.798113	469,043,937	3,752,351,496	86,400
3	6.103643	480,289,911	3,842,319,288	86,400
4	6.196446	400,321,791	3,202,574,328	86,400
5	5.798110	492,939,982	3,943,519,856	86,400
6	3.773516	259,523,954	2,076,191,632	86,400
7	6.103820	448,308,750	3,586,470,000	86,400
8	6.196446	404,095,637	3,232,765,096	86,400
9	6.198637	427,785,452	3,422,283,616	86,400
10	6.196446	406,934,916	3,255,479,328	86,400

Figure 6: Convergence curves of 10 runs by MIDACO on Messenger (full)



value which is less or around $f(x) = 6.2$ (which is still over 200% away from the best known solution).

The refinement run of the best found solution resulted only in a very marginal improved objective function value of 3.772, which is still far away from the best known solution of 1.972.

2.5. Result on GTOP Benchmark Cassini2

The Cassini2 benchmark models an interplanetary space mission to Saturn, including deep space maneuvers (DSM) and is therefore considerable more difficult than its counterpart benchmark: Cassini1. The sequence of fly-by planets for this mission is given by Earth-Venus-Venus-Earth-Jupiter-Saturn, whereas the first item is the start planet and the last item is the final target. The objective of this benchmark to minimize the total deltaV accumulated during the full mission, whereas in contrast to Cassini1 the final deltaV manouvre is considered to be a rendezvous instead of an orbit insection. The benchmark invokes 22 decision variables which are described as follows (for details on hyperbolic trajectories, see Kemble (2006)):

Variable	Description
1	Initial day measured from 1-Jan 2000
2	Initial excess hyperbolic speed (km/sec)
3	Component of excess hyperbolic speed
4	Component of excess hyperbolic speed
$5 \sim 9$	Time interval between events (e.g. departure, fly-by, capture)
$10 \sim 14$	Fraction of the time interval after which DSM occurs
$15 \sim 18$	Radius of flyby (in planet radii)
$19 \sim 22$	Angle measured in planet B plane of the planet approach vector

The currently best known solution published on the ESA Website (2013) was provided by MIDACO in 2009 and corresponds to an objective function value of about $f(x) = 8.3832$. Table 7 displays the numerical results of 10 individual test runs of MIDACO on Cassini2.

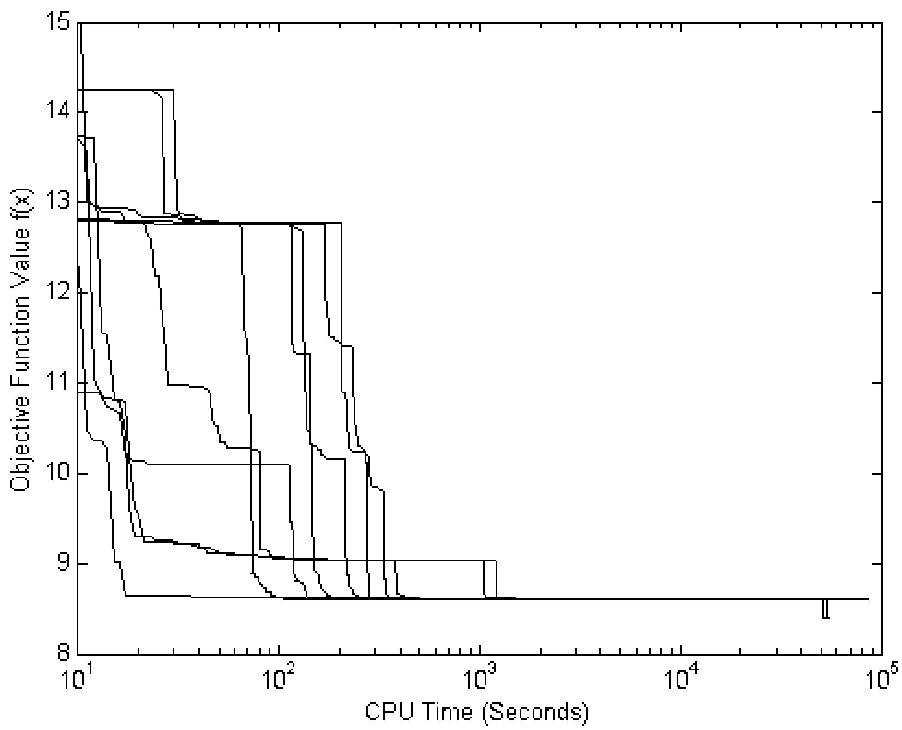
The convergence curves of the 10 individual runs from Table 5 are illustrated in Figure 7. Note that the X-axis in Figure 7 is given in logarithmic scale.

From the results in Table 7 it can be seen that MIDACO is able to reach the best known solution within a precision of 0.1% in 2 out of 10 cases. Test

Table 7: 10 Test runs of MIDACO on Cassini2

Run	$f(x)$	*Blocks*	*Eval*	*Time (Sec)*
1	8.608853	608,948,972	4,871,591,776	86,400
2	8.608904	584,306,388	4,674,451,104	86,400
3	8.608884	619,145,812	4,953,166,496	86,400
4	8.608868	608,206,882	4,865,655,056	86,400
5	8.391325	373,547,088	2,988,376,704	52,029
6	8.391325	373,534,014	2,988,272,112	54,515
7	8.608896	592,815,972	4,742,527,776	86,400
8	8.608861	632,284,426	5,058,275,408	86,400
9	8.608878	604,858,279	4,838,866,232	86,400
10	8.608864	580,064,798	4,640,518,384	86,400

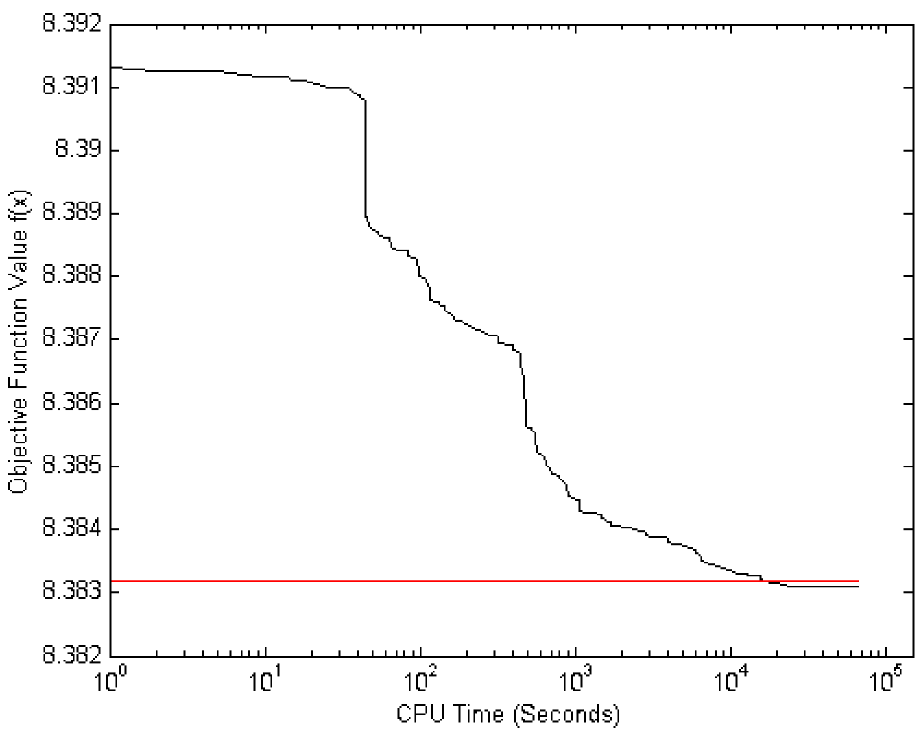
Figure 7: Convergence curves of 10 runs by MIDACO on Cassini2



run number 5 represents the best run corresponding to a cpu-time of 52,029 seconds (about 15 hours) and around 370 Million processed *Blocks*. In all 8 cases were MIDACO does not reach the best known solution within a precision of 0.1%, it obtains an objective function value of around $f(x) = 8.609$, which appears to be a strong local optimum. From Figure 7 it can be seen that after around 1500 seconds all test runs converged to an objective function value around $f(x) = 8.6$ (which is as close as around 2.5% on the current best solution).

Figure 8 illustrates the convergence curve of the refinement run for the solution from test run number 5 of Table 7. In addition to the convergence curve, the value of the best known solution is plotted as a red line in Figure 5. Note that the X-axis in Figure 5 is given in logarithmic scale.

Figure 8: Convergence curves of refinement run on best solution from Table 7



From Figure 8 it can be seen that the exact value of the best known solution is (b)reached at less than 20,000 seconds (less than 6 hours). Furthermore it can be seen from Figure 8 that no further improvements are achieved after around 70,000 seconds. The refined MIDACO solution represents a (very marginal) improvement of about 0.001% in respect to the current best known solution.

2.6. Results on Rosetta

The Rosetta benchmark models multi gravity assist space mission to comet 67P/Churyumov-Gerasimenko, including deep space maneuvers (DSM). The sequence of fly-by planets for this mission is given by Earth-Earth-Mars-Earth-Earth-67P, whereas the first item is the start planet and the last item is the final target. The objective of this benchmark is to minimize the total deltaV accumulated during the full mission. The benchmark invokes 22 decision variables which are described as follows (for details on hyperbolic trajectories, see Kemble (2006)):

Variable	Description
1	Initial day measured from 1-Jan 2000
2	Initial excess hyperbolic speed (km/sec)
3	Component of excess hyperbolic speed
4	Component of excess hyperbolic speed
$5 \sim 9$	Time interval between events (e.g. departure, fly-by, capture)
$10 \sim 14$	Fraction of the time interval after which DSM occurs
$15 \sim 18$	Radius of flyby (in planet radii)
$19 \sim 22$	Angle measured in planet B plane of the planet approach vector

The currently best known solution published on the ESA Website (2013) was provided by the University of Glasgow in 2008 and corresponds to an objective function value of about $f(x) = 1.3433$. Table 8 displays the numerical results of 10 individual test runs of MIDACO on Rosetta.

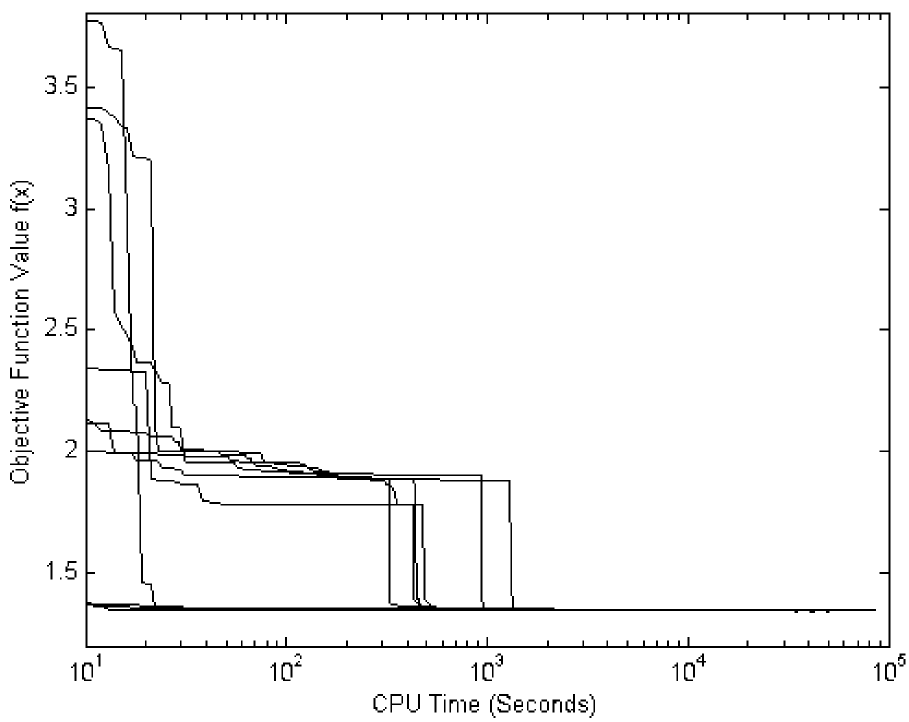
The convergence curves of the 10 individual runs from Table 8 are illustrated in Figure 9. Note that the X-axis in Figure 9 is given in logarithmic scale.

From the results in Table 8 it can be seen that MIDACO is able to reach the best known solution within a precision of 0.1% in 9 out of 10 cases. Test run number 8 represents the best run corresponding to a cpu-time of 583 seconds (less than 10 minutes) and around 4.7 Million processed *Blocks*. In the

Table 8: 10 Test runs of MIDACO on Rosetta

Run	$f(x)$	$Blocks$	$Eval$	$Time\ (Sec)$
1	1.344343	351,389,732	2,811,117,856	50,060
2	1.344317	239,059,215	1,912,473,720	34,617
3	1.344343	155,857,377	1,246,859,016	21,087
4	1.345374	601,973,065	4,815,784,520	86,400
5	1.344295	367,822,959	2,942,583,672	49,784
6	1.344343	314,676,792	2,517,414,336	42,105
7	1.344340	282,036,178	2,256,289,424	40,972
8	1.344343	4,670,118	37,360,944	583
9	1.344338	189,976,320	1,519,810,560	25,890
10	1.344317	238,944,394	1,911,555,152	35,247

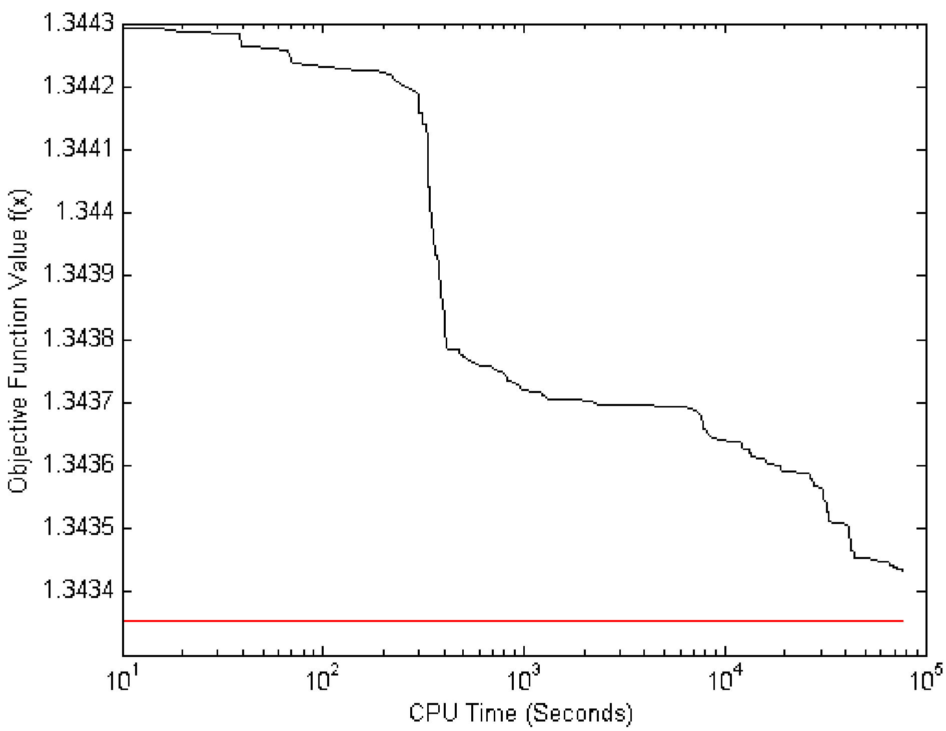
Figure 9: Convergence curves of 10 runs by MIDACO on Rosetta



one case were MIDACO does not reach the best known solution within a precision of 0.1%, it obtains an objective function value of around $f(x) = 1.345$, which is still close to the best known solution. From Figure 9 it can be seen that after around 1500 seconds all test runs converged to an objective function value less than $f(x) = 1.4$ (which is closer than 5% on the best known solution).

Figure 10 illustrates the convergence curve of the refinement run for the solution from test run number 5 of Table 8. In addition to the convergence curve, the value of the best known solution is plotted as a red line in Figure 10. Note that the X-axis in Figure 10 is given in logarithmic scale.

Figure 10: Convergence curves of refinement run on best solution from Table 8



From Figure 10 it can be seen that the exact value of the best known solution is not reached within the cpu-time budget of 24 hours; instead, after the full cpu-time budget is spent the refined solution remains around 0.01% above the best known solution.

2.7. Results on Sagas

The Sagas benchmark models is described as a deltaV-EGA manouvre to fly-by Jupiter and reach the keplerian orbit of 50AU. The sequence of fly-by planets for this mission is given by Earth-Earth-Jupiter, whereas the first item is the start planet and the last item is the final target. The objective of this benchmark is to minimize the total deltaV accumulated during the full mission. The benchmark invokes 12 decision variables which are described as follows (for details on hyperbolic trajectories, see Kemble (2006)):

Variable	Description
1	Initial day measured from 1-Jan 2000
2	Initial excess hyperbolic speed (km/sec)
3	Component of excess hyperbolic speed
4	Component of excess hyperbolic speed
$5 \sim 6$	Time interval between events (e.g. departure, fly-by, capture)
$7 \sim 8$	Fraction of the time interval after which DSM occurs
$9 \sim 10$	Radius of flyby (in planet radii)
$11 \sim 12$	Angle measured in planet B plane of the planet approach vector

This benchmark further considers two constraints, which impose an upper limit on the on-board fuel and launcher performance. The currently best known solution published on the ESA Website (2013) was provided by a differential evolution algorithm in 2005 and corresponds to an objective function value of about $f(x) = 18.1936$. This solution remains the only submission to this benchmark, it can therefore be considered a relatively easy one. Table 9 displays the numerical results of 10 individual test runs of MIDACO on Sagas.

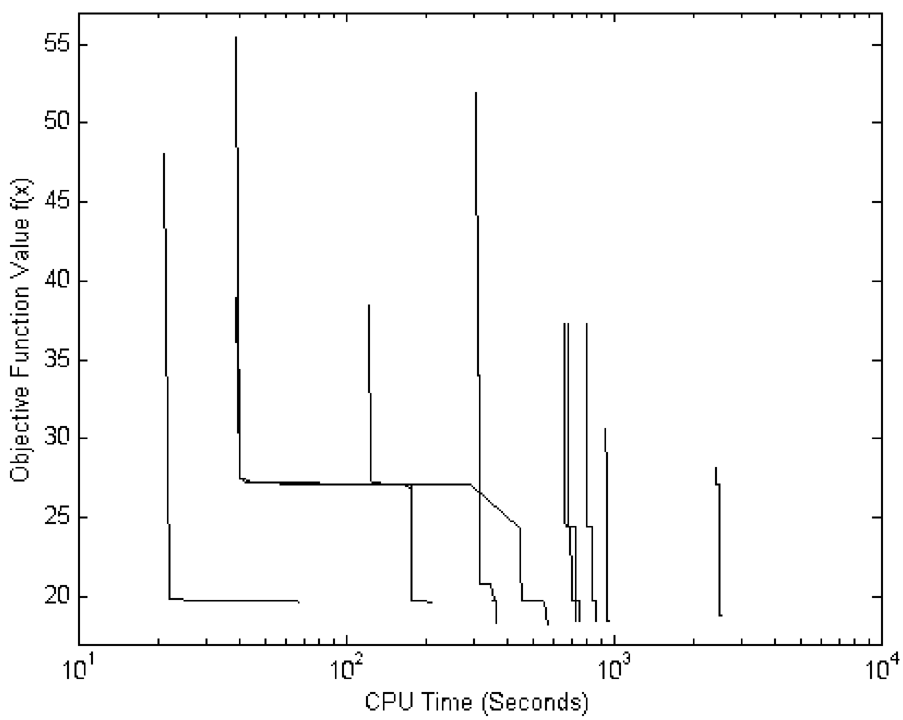
The convergence curves of the 10 individual runs from Table 9 are illustrated in Figure 11. Note that the X-axis in Figure 11 is given in logarithmic scale. Further note that Figure 11 only displays the convergence curves of feasible solutions with an objective function value less than 100. Values over 100 are discarded for better readability.

From the results in Table 9 it can be seen that MIDACO is able to reach the best known solution within a precision of 0.1% in all cases. Test run number 1 represents the best run corresponding to a cpu-time of 67 seconds

Table 9: 10 Test runs of MIDACO on Sagas

Run	$f(x)$	*Blocks*	*Eval*	*Time (Sec)*
1	18.203876	1,597,711	12,781,688	67
2	18.207551	10,072,832	80,582,656	365
3	18.207517	4,432,026	35,456,208	172
4	18.207866	20,239,855	161,918,840	724
5	18.208116	5,292,229	42,337,832	210
6	18.207517	23,777,511	190,220,088	964
7	18.207866	20,236,698	161,893,584	750
8	18.207866	20,216,963	161,735,704	862
9	18.208087	12,887,894	103,103,152	565
10	18.208154	54,127,623	433,020,984	2517

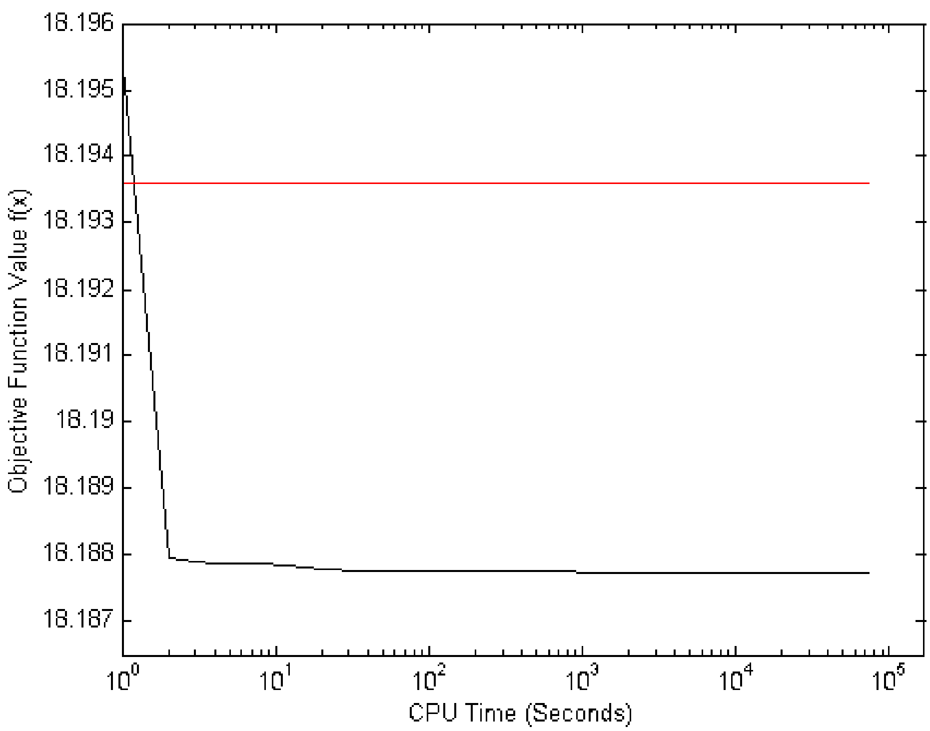
Figure 11: Convergence curves of 10 runs by MIDACO on Sagas



and 1,597,711 processed *Blocks*. Test run number 10 represents the worst run corresponding to a cpu-time of 2,517 seconds and 54,127,623 processed *Blocks*. From Figure 11 it can be seen that (except for test run number 10) after around 1,000 seconds all test runs converged to an objective function value less than $f(x) = 20$ (which is closer than 10% on the best known solution).

Figure 12 illustrates the convergence curve of the refinement run for the solution from test run number 1 of Table 9. In addition to the convergence curve, the value of the best known solution is plotted as a red line in Figure 12. Note that the X-axis in Figure 12 is given in logarithmic scale.

Figure 12: Convergence curves of refinement run on best solution from Table 9



From Figure 12 it can be seen that the exact value of the best known solution is (b)reached already below 2 seconds. Furthermore it can be seen from Figure 12 that a significant improvement on the precision is achieved. The refined MIDACO solution represents an improvement of about 0.04% in respect to the current best known solution. The constraint violation of the solution is zero.

3. Summary

This section gives an overview on the numerical results obtained in Section 2 for all considered benchmarks. In regard to an ***out-of-the-box*** approach, MIDACO was able to obtain the best known solution for 5 out of 7 benchmarks within a precision of 0.1%. The two instances in which MIDACO did not reach the best known solution were GTOC1 and Messenger (full). As Messenger (full) appears to be considerable more difficult than any of the other benchmarks, it is not surprising that MIDACO failed to solve this problem within the given time limit of 24 hours. In contrast to Messenger (full), MIDACO's failure on GTOC1 is likely to be caused by the reduced cpu-time budget of only 2.5 hours instead of 24 hours.

Table 10 gives a summary of the numerical results by MIDACO from Section 2. The (approximate) cpu-time to reach the current best known solution within a precision of 0.1% is reported for the best test run and as the average over all successful test runs. Furthermore, Table 10 reports the success rate out of the 10 individual runs.

Table 10: Summary of numerical results from Section 2 for all GTOP benchmarks

Benchmark	Best Time	Average Time	Success Rate
Cassini1	10 seconds	1 minute	100%
GTOC1	-	-	0%
Messenger (reduced)	1 hour	9 hours	60%
Messenger (full)	-	-	0%
Cassini2	15 hours	15 hours	20%
Rosetta	10 minutes	9 hours	90%
Sagas	1 minute	12 minutes	100%

4. Conclusions & Future Work

With currently 3 unbroken record solutions, including the one for the most difficult instance, the MIDACO software has proven its potential on the interplanetary trajectory benchmarks from the GTOP database in the past. Here a rigorous numerical evaluation of the ***out-of-the-box*** performance of MIDACO on that database was presented. The GTOP benchmarks are known to be difficult and previous solution submission times for those problems span several month and even years (see Table 1). In Section 2 it was shown that MIDACO is able to solve five out of seven of these problems to their best known solution within minutes to hours (see Table 10). As, to the best knowledge of the authors, such an out-of-the-box performance on the GTOP database is yet unmatched by any other available optimization software, it is concluded that MIDACO currently represents the state-of-the-art for interplanetary space mission trajectory optimization.

The presented numerical results were obtained under exploitation of MIDACO's parallelization feature for multi-core CPU's. The here considered parallelization factor of 8 was set for a hyper-threaded quad-core cpu and remains relatively low in the context of currently available parallel architectures. Future research might investigate the impact of significantly higher parallelization factors on massively parallelizable computer architectures. Further reductions of the cpu-time required to solve this kind of application are expected by such an approach, as it was recently indicated in (see Schlueter & Munetomo (2014).

Acknowledgement

I would like to express my gratitude to D. Izzo and the Advanced Concept Team (ACT) for creating and publishing the GTOP benchmark set. I would also like to thank the European Space Agency (ESA-ESTEC/Contract No. 21943/08/NL/ST) and EADS Astrium Ltd (Stevenage, UK) for their professional (in particular S. Kemble, S. Erb and G. Ortega) and financial support on the MIDACO development. Furthermore I would like to thank Prof. M. Munetomo for valuable comments and providing the computer equipment for the presented results. Finally, I am thankful to the JASR reviewers for their contribution of technical details on the space mission benchmarks and an overall improved presentation.

References

Addis, B., Cassioli, A., Locatelli, M. & Schoen, F., Global optimization for the design of space trajectories. Comput. Optim. Appl. 48(3), 635-652, 2011.

Ampatzis, C., & Izzo, D.: Machine learning techniques for approximation of objective functions in trajectory optimisation, Proc. Int. Conf. Artificial Intelligence in Space (IJCAI), 2009.

Biazzini, M., Banhelyi, B., Montresor, A. & Jelasity, M: Distributed Hyperheuristics for Real Parameter Optimization, Proc. 11th Ann. Conf. Genetic and Evolutionary Computation (GECCO), 1339-1346, 2009.

Biscani, F., Izzo, D. & Yam, C.H.: A Global Optimisation Toolbox for Massively Parallel Engineering Optimisation, Proc. 4th Int. Conf. Astrodynamics Tools and Techniques (ICATT), 2010.

Danoy, G., Pinto, F.,G., Dorronsoro, B. & Bouvry, P.: New State-Of-The-Art Results For Cassini2 Global Trajectory Optimization Problem, Acta Futura 5, 65-72, 2012.

European Space Agency (ESA) and Advanced Concepts Team (ACT). Gtop database - global optimisation trajectory problems and solutions, Software available at http://www.esa.int/gsp/ACT/inf/projects/gtop/gtop.html, 2013.

Gad, A.,H.,G.,E.: Space trajectories optimization using variable-chromosome-length genetic algorithms, PhD-Thesis, Michigan Technological University, USA, 2011.

Gruber, A.: Multi Gravity Assist Optimierung mittels Evolutionsstrategien, BSc-Thesis, Vienna University of Technology, Austria, 2009.

Henderson, T.,A.: A Learning Approach To Sampling Optimization: Applications in Astrodynamics, PhD-Theis, Texas A & M University, USA, 2013.

Islam, S.K.M., Roy, S.G.S. & Suganthan, P.N.: An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization, IEEE Transactions on Systems, Man and Cybernetics 42(2), 482-500, 2012

Izzo, D.: 1st ACT global trajectory optimisation competition: Problem description and summary of the results Acta Astronaut. 61(9), 731-734, 2007.

Izzo, D.: Global Optimization and Space Pruning for Spacecraft Trajectory Design. *Spacecraft Trajectory Optimization* Conway, B. (Eds.), Cambridge University Press, 178-199, 2010.

Kemble, S.: Interplanetary Mission Analysis and Design. *Astronautical Engineering*, Springer , 2006.

Lancinskas, A., Zilinskas, J. & Ortigosa, P., M.: Investigation of parallel particle swarm optimization algorithm with reduction of the search area, Proc. Int. Conf. Cluster Computing Workshops and Posters (IEEE), 2010.

Musegaas, P.: Optimization of Space Trajectories Including Multiple Gravity Assists and Deep Space Maneuvers, *MSc Thesis*, Delft University of Technology, Netherlands, 2012.

Schlueter, M., Egea, J.A. & Banga, J.R.: Extended ant colony optimization for non-convex mixed integer nonlinear programming, Comput. Oper Res. 36(7), 2217-2229, 2009.

Schlueter, M. & Gerdts., M.: The oracle penalty method. J. Global Optim. 47(2), 293-325, 2010.

Schlueter, M., Gerdts, M. & Rueckmann, J.J.: A numerical study of MIDACO on 100 MINLP benchmarks, Optimization 61(7), 873-900, 2012a.

Schlueter, M.: Nonlinear Mixed Integer Based Optimisation Technique for Space Applications, PhD Thesis, School of Mathematics, University of Birmingham, United Kingdom, 2012b.

Schlueter M., Erb S., Gerdts M., Kemble S., & Rueckmann J.J.: MIDACO on MINLP Space Applications. Advances in Space Research, 51(7), 1116-1131, 2013a.

Schlueter M. & Munetomo M.: Parallelization Strategies for Evolutionary Algorithms for MINLP. Proc. Int. Cong. Evolutionary Computation (IEEE-CEC), 635-641, 2013b.

Schlueter M. & Munetomo M.: Parallelization for Space Trajectory Optimization. Proc. Int. World Cong. Computational Intelligence (IEEE-WCCI), *accepted*, 2014.

Schlueter M. & Munetomo M.: Introduction to MIDACO-SOLVER Software. Technical Report, HUSCAP, Hokkaido University, Japan, 2013c.

Stracquadanio, G., La Ferla, A., De Felice, M. & Nicosia, G.: Design of robust space trajectories, Proc. 31st Int. Conf. Artificial Intelligence (SGAI), 2011.

Vinko, T. & Izzo, D.: Global Optimisation Heuristics and Test Problems for Preliminary Spacecraft Trajectory Design, European Space Agency, ACT Tec. Rept. ACT-TNT-MAD-GOHTPPSTD, 2008.