A Mixed-Integer Extension for ESA's Cassini1 Space Mission Benchmark

Martin Schlueter
Information Initiative Center
Hokkaido University
Sapporo 060-0811, Japan
schlueter@midaco-solver.com

Masaharu Munetomo
Information Initiative Center
Hokkaido University
Sapporo 060-0811, Japan
munetomo@iic.hokudai.ac.jp

***Abstract*—This contribution introduces a mixed-integer extension to the well-known Cassini1 space mission benchmark published by the European Space Agency (ESA). Due to it's highly nonlinear function properties, the Cassini1 benchmark is widely recognized by the evolutionary computing community as interesting test case. The Cassini1 benchmark implements a simplified model of an interplanetary space trajectory from Earth to Saturn, using four gravity-assist maneuvers (also known as *fly-by's*) at planets: Mercury, Mercury, Earth and Jupiter. Here the original continuous formulation of the Cassini1 benchmark is extended by four discrete (integer) variables, representing the choice of *fly-by* planets. Comprehensive numerical results investigate the complexity of this mixed-integer formulation and compare it with the purely continuous formulation. The results show that the difficulty to solve the mixed-integer formulation can significantly vary depending on small modifications of the integer search space. Preliminary multi-objective results are additionally presented in order to deepen the understanding of the observed mixed-integer performance.**

***Index Terms*—MINLP, Ant Colony Optimization, MIDACO**

I. Introduction

The design of interplanetary space flight trajectories remains a challenging and active area for applying global optimization algorithms. Since 2005 the Advanced Concept Team (ACT) of the European Space Agency (ESA) publishes a database of Global Trajectory Optimization (GTOP) benchmarks [6]. All GTOP benchmarks are originally formulated as single-objective optimization problems. The easiest[1] and most widely used instance of the GTOP set is the *Cassini1* benchmark problem, which consist of six continuous decision variables and four non-linear constraints. An overview on previous attempts to solve the GTOP problems can be found in [6]. This contribution presents a mixed-integer[2] extension to the original formulation by adding four additional discrete (also called integer or combinatorial) optimization variables. Those four variables represent the sequence of *fly-by* planets, where each of the nine[3] planets of the solar system are a possible choice.

[1]In contrast to Cassini1, the Messenger (full version) [14] benchmark is considered to be the most difficult instance of the GTOP benchmark set.

[2]To the authors best knowledge, this is the first time that the Cassini1 benchmark is considered as mixed-integer problem.

[3]Pluto is considered the ninth-planet here.

Figure 1 illustrates the original Cassini space probe trajectory launched in 1997 by NASA. The original sequence of *fly-by* planets was Venus (Apr-1998), Venus (Jun-1999), Earth (Aug-1999) and Jupiter (Dec-2000). Those *fly-by*'s are observable from Figure 1.

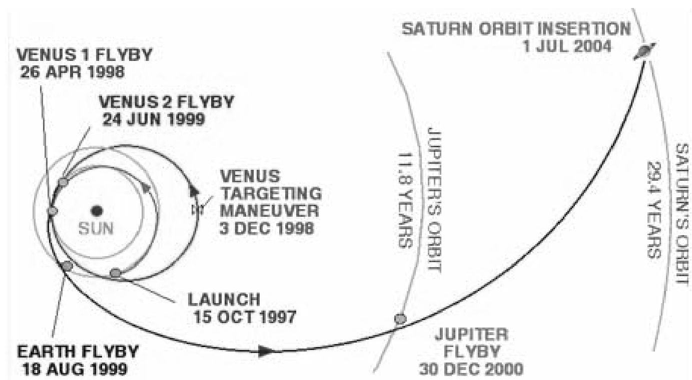

Fig. 1. Interplanetary trajectory of NASA's Cassini Mission (1997)

Extending the original formulation by four discrete variables exponentially increases the variable search space and allows many more possible trajectories to be explored and evaluated. Mixed-integer formulations of interplanetary trajectories are known to be difficult to solve and remain a rare exception in the literature on trajectory optimization, see e.g. [5], [8], [9] [12] or [18] .

This contribution is focused on estimating and comparing the complexity of the mixed-integer formulation with the original purely continuous formulation. Extensive numerical results with the MIDACO optimization software [12] are presented for both formulations, revealing that the mixed-integer formulation can be significantly more difficult to solve. Furthermore, several local optimal mixed-integer solutions are investigated individually; revealing a challenging complexity of the objective search space with one particular strong and decisive local optimum with planet sequence Earth-Earth-Earth-Jupiter.

Additionally, preliminary numerical results for a multi-objective mixed-integer formulation of Cassini1 are presented. Those results enable a graphically illustration of the objective space, which provides a deeper understanding of the nature and complexity of the above mentioned mixed-integer local optimal solutions.

A conclusion section finally summarizes the numerical results and gives an outlook on further research.

II. Mixed-Integer Extension for Cassini1

This section briefly describes the details of the considered mixed-integer extension of the Cassini1 benchmark. Table I lists the total number of continuous variables, whereas the four last variables (x_7, x_8, x_9, x_{10}) are of integer type and represent the sequence of *fly-by* planets.

TABLE I
Mixed-Integer Optimization Variables

Variable	Description	Type
x_1	Launch date (MJD)	continuous
x_2, x_3, x_4, x_5, x_6	Days between events	continuous
x_7, x_8, x_9, x_{10}	*fly-by* planet sequence	discrete

Table II list all nine planets with their corresponding integer value, that is used as discrete input variable for the mixed-integer formulation.

TABLE II
Possible choice of *fly-by* Planets

Value	Planet
1	Mercury
2	Venus
3	Earth
4	Mars
5	Jupiter
6	Saturn
7	Uranus
8	Neptune
9	Pluto

The technical modification in the source code of the GTOP database, necessary to enable the integer choice of *fly-by* planets, is given in the file `trajobjfunc.cpp` . In such file, the following original source code line:

```
sequence_[CASSINI_DIM] = {3,2,2,3,5,6}
```

is replaced with the following:

```
sequence_[CASSINI_DIM] = {3,x7,x8,x9,x10,6}
```

Whereas the sequence elements x_7,x_8,x_9,x_{10} represent the integer variables. Note that the first and last entry in the above source code sequence represent the start and target planet, which is in this case Earth ($\rightarrow$ start) and Saturn ($\rightarrow$ target). Therefore the first and last entry in the original sequence remain unaffected by the above modification.

III. MIDACO Algorithm

This section gives a brief introduction into the underlying mixed integer optimization algorithm implemented in the MIDACO optimization software [12]. MIDACO is based on an evolutionary computing method known as *Ant Colony Optimization* (ACO), see e.g. [4], [1] or [7]. ACO is a heuristic method that aims to approximate good and potential global optimal solutions to problems that are formulated as *black-box* problem, thus no *white-box* (inside) knowledge of the optimization problem is required to apply ACO.

For mixed integer search domains, MIDACO uses an ACO extension described in full detail in [9]. This ACO extension differs in the way the ACO algorithm creates it's iterates (also called *ants*) based on the previously gained solution knowledge (also called *pheromones*). While in case of continuous variables a continuous multi-kernel Gauss probability density distribution (PDF) is applied (see Figure 2), a discretized version is used for the integer search space (see Figure 3).

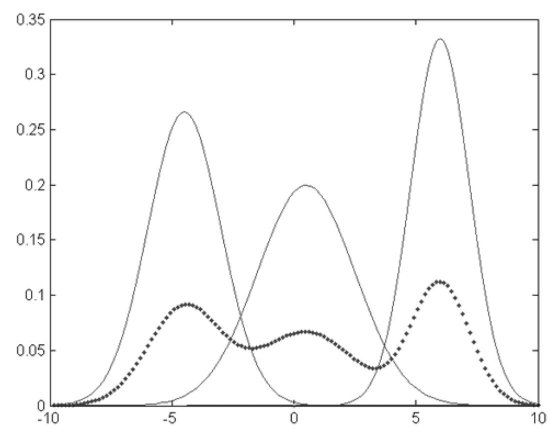

Fig. 2. Continuous multi-kernel PDF used within ACO algorithm

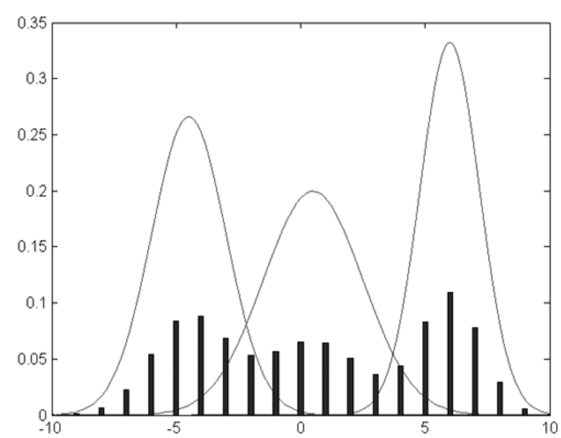

Fig. 3. Discretized multi-kernel PDF used within ACO algorithm

A comprehensive numerical study on the effectiveness of MIDACO on 200 mixed-integer benchmark problems can be found in [13], showing that the algorithm is capable to solve the majority of benchmarks to their global solution in reasonable time.

The MIDACO algorithm has further been extensively tested on interplanetary space trajectory design problems [12] and holds several record solutions on the GTOP database, including the best known solution to the Messenger (full version) instance [14], which is considered the most difficult instance of the database.

Constraints are handled within MIDACO via the oracle penalty method [11], which is a self-adaptive method specifically developed for evolutionary and heuristic algorithms. MIDACO is used with its default algorithmic parameters for all results presented in the numerical section.

IV. Numerical Results

This section presents numerical results on the mixed-integer extended version of the Cassini1 benchmark [6]. This section is divided into two subsections, the first one considers Cassini1 as single-objective problem while the second one considers multiple objectives. The MIDACO version 6.0 optimization software [12] was used to solve the optimization problem. All numerical experiments where conducted on an Intel(R) Xeon(R) CPU E5-2650L v3 @ 1.80GHz

A. Single-Objective Mixed-Integer Results

This section presents numerical results on a single-objective formulation of the mixed-integer extended Cassini1 benchmark. The first goal of this section is to identify the individual best global optimal solutions corresponding to different planet sequences, of which there are $9^4 = 6561$ possible combinations in total. The second goal of this section is to compare the numerical complexity to solve the continuous (denoted as NLP) formulation with the mixed-integer (denoted as MINLP) formulation.

In order to explore the mixed-integer search space and identify the ten best global solutions for different planet sequences, the MIDACO optimization software was applied repeatedly ten times for a fixed CPU time budget of 10,000 seconds (roughly 3 hours[1]) on the mixed-integer formulation, whereas all integer sequences from previous found best known solutions where excluded as feasible solution from any further runs. As a result, Table III lists the different planet sequences for the ten best known solutions together with their corresponding objective function value (ΔV). Note that the original planet sequence (2,2,3,5) appears as the fourth best solution, corresponding to the well known optimum of $\Delta V = 4.9307$. The fact that three solutions with different planet sequence and better objective value were identified is likely due to the highly simplified model of the Cassini1 benchmark.

In the further proceeding of this section, we will focus only on the first four best solutions, from which the original sequence (2,2,3,5) holds the worst objective function value

[1]10,000 seconds for each individual test run

TABLE III
Best 10 integer combinations out of $9^4 = 6561$ possible

ΔV	Solutions Fly-Bye Planet Sequence			
3.5007	3 (Earth)	2 (Venus)	3 (Earth)	5 (Jupiter)
3.6307	3 (Earth)	3 (Earth)	3 (Earth)	5 (Jupiter)
4.7556	3 (Earth)	3 (Earth)	3 (Earth)	6 (Saturn)
4.9307	**2** (Venus)	**2** (Venus)	**3** (Earth)	**5** (Jupiter)
5.0177	3 (Earth)	3 (Earth)	2 (Venus)	5 (Jupiter)
5.1631	3 (Earth)	2 (Venus)	2 (Venus)	5 (Jupiter)
6.9289	3 (Earth)	2 (Venus)	2 (Venus)	6 (Saturn)
6.2201	3 (Earth)	2 (Venus)	3 (Earth)	6 (Saturn)
7.7364	2 (Venus)	2 (Venus)	3 (Earth)	6 (Saturn)
9.0582	4 (Mars)	4 (Mars)	4 (Mars)	5 (Jupiter)

($\Delta V = 4.9307$) in comparison. Table IV lists the full solution values for the four best identified integer sequences. Note that all solutions significantly differ from each other in the continuous variables. In regard to the integer sequence, those four solutions agree only in such regard, that the 3rd fly-by planet of the sequence is Earth.

TABLE IV
Best 4 individual solutions with different sequence

ΔV :	3.5007	3.6307	4.7556	**4.9307**
x_1 :	-768.507	-836.986	-825.243	-789.766
x_2 :	350.595	207.913	171.311	158.316
x_3 :	234.191	192.809	235.345	449.385
x_4 :	55.791	258.400	250.770	54.709
x_5 :	1012.700	910.784	1999.999	1024.763
x_6 :	4533.926	4405.162	5505.540	4552.914
x_7 :	3 (Earth)	3 (Earth)	3 (Earth)	**2** (Venus)
x_8 :	2 (Venus)	3 (Earth)	3 (Earth)	**2** (Venus)
x_9 :	3 (Earth)	3 (Earth)	3 (Earth)	**3** (Earth)
x_{10} :	5 (Jupiter)	5 (Jupiter)	6 (Saturn)	**5** (Jupiter)

In order to estimate the complexity of the mixed-integer formulation of the Cassini1 benchmark, a comprehensive study of the purely continuous NLP case is conducted first. This is done by fixing the planet sequence for the best four cases (see Table IV), thus creating four different instances of a purely continuous formulation. Table V shows the numerical results of 30 independent test runs with MIDACO on each of such four instances. Table V lists the best, worst and mean number of required function evaluation and corresponding CPU-time, to reach the individual best solution with a precision of 0.1%[1].

From Table V it is interesting to see that the complexity of the continuous formulation may drastically vary due to the considered planet sequence. The planet sequence (3,3,3,6) creates apparently the most difficult instance, requiring 182.4 seconds on average to be solved. In contrast, the sequence

[1]The official precision by the GTOP database is 0.1% for any new solution to be included into the database.

TABLE V
COMPUTATIONAL EFFORT TO SOLVE FIRST FOUR NLP SETUPS

	Evaluation			CPU time [sec]		
Seq.	Best	Worst	**Mean**	Best	Worst	**Mean**
3 2 3 5	648,852	10.1E+6	3.8E+6	11.8	179.2	70.3
3 3 3 5	21,697	1.3E+6	156,479	0.4	22.2	2.7
3 3 3 6	1.1E+6	83.2E+6	10.3E+6	18.8	1465.2	182.4
2 2 3 5	326,003	2.7E+6	1.3E+6	5.4	44.8	21.8

(3,3,3,5) creates apparently the easiest instance, requiring only 2.7 seconds on average to be solved. The original sequence (2,2,3,5) appears moderately easy, with 21.8 seconds on average to be solved.

As a next step, the mixed-integer formulation is considered. Here, four different MINLP scenarios are created, which only differ in such sense, that for some instances, some sequences are artificially made infeasible (by using an additional constraint). This is done in order to investigate in detail the MINLP cases corresponding to the four best sequences and later on compare their complexity with the continuous NLP cases. The following MINLP scenarios are considered:

Scenario A: all sequences are feasible
Scenario B: only sequence (3,2,3,5) is infeasible
Scenario C: only sequence (3,3,3,5) is infeasible
Scenario D: sequence (3,3,3,5) and (3,2,3,5) are infeasible
Scenario E: sequence (3,3,3,5),(3,2,3,5),(3,3,3,6) are infeasible

Table VII lists the best, worst and mean[1] number of required function evaluation and corresponding CPU-time, to reach the individual best solution with a precision of 0.1%. A maximal CPU time budget of 10,000 seconds ($\sim$ 3 hours) was applied to each individual test run. Table VI shows the success rate for each of the five MINLP scenarios[2]. It is important to note that for the scenario A special behavior is observed: For scenario A the MIDACO algorithm quickly converges to the second best integer sequence of (3,3,3,5) in all test runs, rather than to converging to the apparent global optimal solution with sequence (3,2,3,5). Therefore the success rate in reaching the global optimal solution is zero for scenario A. This observation indicates that reaching the global MINLP solution with sequence (3,2,3,5) is significantly more difficult than anticipated. The following Subsection IV-B will further investigate this issue and give an explanation for this behavior.

From Table VII it can be seen that the complexity of the five considered MINLP instances drastically varies. While the scenario B MINLP is solved within 31.3 seconds on average, the scenarios C and E require on average well over an hour

[1]Note that the reported mean value underestimates the actual mean value to reach the threshold of 0.1% in cases where the run was stopped due to the maximal time budget.

[2]Note that the sequence (3,2,3,5) was found (and presented in Table III) by modifying the search space so that sequence (3,3,3,5) was made infeasible. This procedure is therefore equal to Scenario C, described in above schema. This means that when sequence (3,3,3,5) is made infeasible, the algorithm had a chance of 70% to reach the supposed global solution of ΔV=3.5007, while it had a zero chance to find this solution, if all sequences were feasible.

TABLE VI
SUCCESS RATE FOR FOUR MINLP SCENARIOS

Scenario	Success Rate	
A	0%	(0 out of 30)
B	100%	(30 out of 30)
C	70%	(21 out of 30)
D	97%	(29 out of 30)
E	57%	(17 out of 30)

TABLE VII
COMPUTATIONAL EFFORT TO SOLVE FIRST FOUR MINLP SCENARIOS

	Evaluation			CPU time [sec]		
Scen.	Best	Worst	**Mean**	Best	Worst	**Mean**
A	-	-	174.0E+6	-	-	10000.0
B	203,015	4.6E+6	1.1E+6	5.5	127.5	31.3
C	1.5E+6	268.0E+6	173.9E+6	42.1	10000.0	4770.9
D	2.0E+6	381.2E+6	32.7E+6	53.9	10000.0	925.9
E	4.2E+6	371.8E+6	172.3E+6	114.9	10000.0	4834.7

to be solved. This different complexity is also reflected in the success rate, which is very high with 100% and 97% for scenarios B and D, while scenario C and E had a success rate of only 70% and 57% respectively.

With the data from Table V and Table VII it is now possible to compare the complexity of the four considered NLP and MINLP scenarios. Table VIII shows the estimated complexity difference between the varying NLP and MINLP scenarios, whereas the ">" symbol means "more than" and the ">>" means "significantly more than". The estimation is based on comparing the best, worst and mean number of function evaluation between the NLP and the MINLP cases.

For example: In case of scenario D (which corresponds to a best known sequence of (3,3,3,6)) the mean number of NLP evaluation where 10322225 and the mean number of MINLP evaluation was 32770441 (with a success rate of 97%). Therefore the mean complexity difference for scenario D is calculated as $\frac{32770441}{10322225} \approx 3.17$ and denoted as > 3.17, because the success rate was just 97% rather than 100% for the MINLP case.

TABLE VIII
COMPLEXITY COMPARISON BETWEEN NLP AND MINLP SETUPS

Scenario	Sequence	Best	Worst	**Mean**
A	(3,2,3,5)	∞	∞	∞
B	(3,3,3,5)	9.35	3.48	7.25
C	(3,2,3,5)	2.4	$>>$ 116.55	$>>$ 65.21
D	(3,3,3,6)	1.79	$>$ 4.57	$>$ 3.17
E	(2,2,3,5)	13.02	$>>$ 136.51	$>>$ 130.79

From Table VIII it can be seen that the complexity can significantly vary between the different considered scenarios, being as low as > 3.17 and 7.25 for scenario B and D. It can be seen that scenario C and E have a significantly higher complexity with $>>$ 65.21 and $>>$ 130.79 respectively. Scenario A represents the highest possible complexity gain with "∞", because the MINLP version of scenario A was never solved to its global optimum (see Table VII) in any of the conducted test runs in this study.

B. Multi-Objective Space Mapping for Mixed-Integer Cases

This section presents some preliminary numerical results for a multi-objective formulation of the mixed-integer combinations of the Cassini1 benchmark described in Section II. The multi-objective extension is described in [15]. Two multi-objective scenarios are considered here: The first one assumes only two objectives, which are the total ΔV as first objective $f_1(x)$ and the total flight time as second objective $f_2(x)$. The second one assumes four objectives which are listed in Table IX.

TABLE IX
DESCRIPTION OF FOUR OBJECTIVES FOR CASSINI1

Objective	Description	Unit
$f_1(x)$	Total ΔV (including ΔV_∞)	km/sec
$f_2(x)$	Time of Flight	Days
$f_3(x)$	Launch Date	MJD2000
$f_4(x)$	Launch ΔV_∞	km/sec

Purpose of this section is to create a detailed visual mapping of the objective space in regard to the MINLP scenarios described in Section Section IV-A. Such mapping should then be investigated in order to better understand the varying complexity of the MINLP scenarios (see Table VIII) and particularly the complexity of scenario A, which had a zero success rate to reach the global optimum (see Table VI).

The MIDACO optimization software has been applied with a fixed CPU-time budget of **24 hours**[1] on each of the four MINLP scenarios B, C, D and E for the two multi-objective scenarios. Therefore the computational resources for the results in this section required eight days in total. However, note that these results are not primarily concerned with optimization performance, but with creating a most complete mapping of the objective space, which is a notoriously intensive task for hard problems.

Figure 4 shows the results for the two-objective case with a first objective function value below 6.0. Figure 5 displays a close-up view on the most interesting part of Figure 4, which is an intersection of the pareto front corresponding to the *fly-by* planet sequence of (3,2,3,5) with the pareto front corresponding to the sequence of (3,3,3,5).

As hoped, the results displayed in Figure 4 and Figure 5 do indeed give an indication why the MINLP scenario A is so much harder to solve than any of the other considered MINLP scenarios. This can be observed by the pareto fronts corresponding to the solutions with sequence (3,2,3,5) (colored grey) and sequence (3,3,3,5) (colored orange). As it can be seen from Figure 4 the orange pareto front is positioned well "below" the grey pareto front for the most part, where "below" refers here to the position regarding both: the x-axis (expressing the first objective, total ΔV) and the y-axis (expressing the second objective, flight time). Only for solutions with a flight time above 5500 days, the grey pareto front holds a better first objective value, than the orange pareto front. This critical part is displayed as close up in Figure 5, where the intersection between the grey and orange pareto front is well observable. The global optimum to the MINLP scenario A is located at the extremal point of the grey pareto front.

Because the orange pareto front significantly dominates the grey pareto front for the most part (in especially for any flight time below 5500 days), it becomes apparent why the optimization algorithm, when applied on the MINLP scenario A (which allows all possible sequences as feasible) gets attracted much quicker to a solution with sequence (3,3,3,5) rather than (3,2,3,5). In other words: The objective space

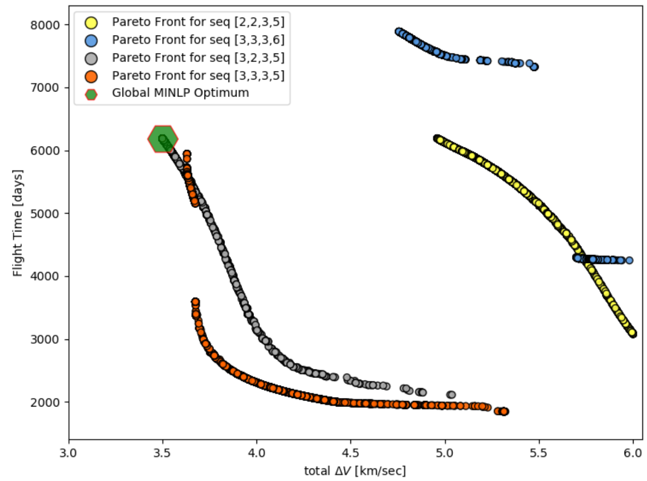

Fig. 4. Two-Objective Space for solutions with $f_1(x) < 6$

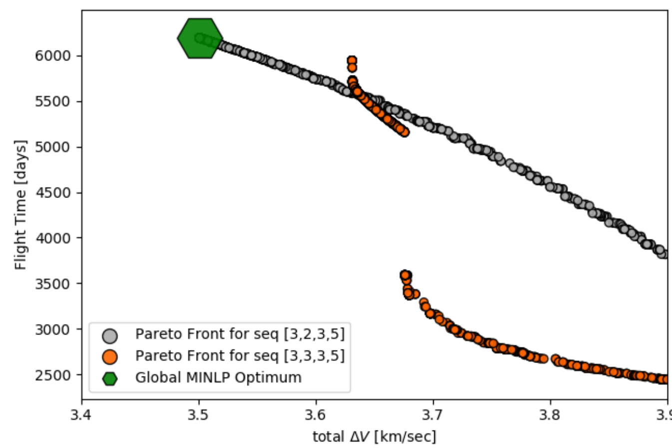

Fig. 5. Two-Objective Space for solutions with $f_1(x) < 3.9$

[1] Corresponding roughly to some thousand million evaluation.

mapping indicates that solutions with sequence (3,3,3,5) pose a very decisive sub-optimum in MINLP scenario A case, leading the optimization algorithm to the second best solution (which has an $f_1(x) = 3.6307$) rather than to the global optimum of $f_1(x) = 3.507$.

The observed behavior of pareto fronts for the two-objective case is now further investigated under the four-objective scenario. Figure 6 shows the results for the four-objective case with a first objective function value below 30.0. Figure 7 displays a close-up view of Figure 6 of solutions with a first objective function value below 6.0.

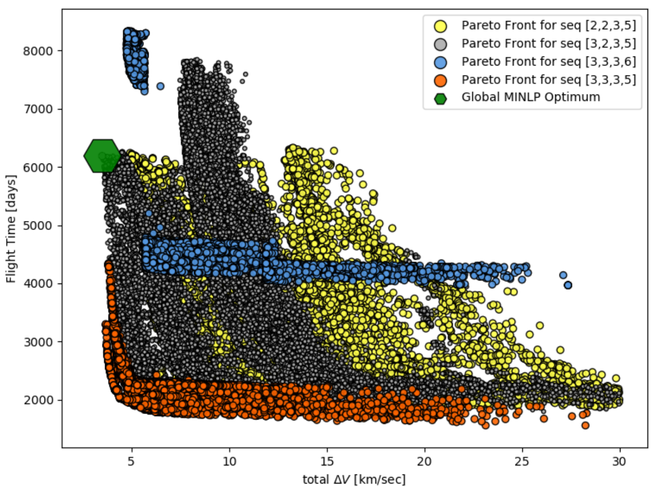

Fig. 6. Four-Objective Space for solutions with $f_1(x) < 30$

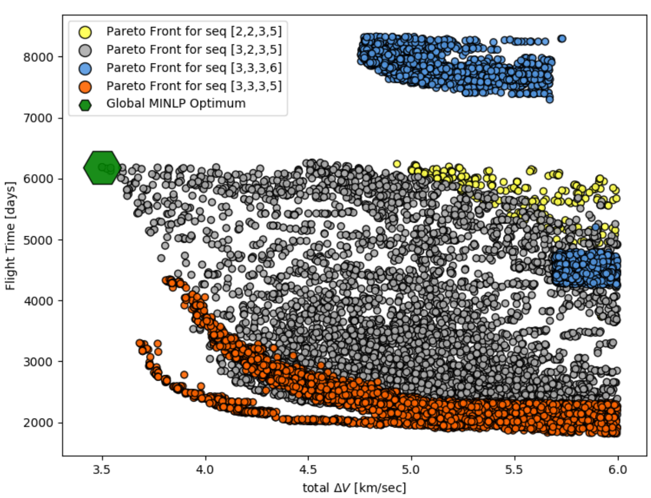

Fig. 7. Four-Objective Space for solutions with $f_1(x) < 6$

Again, it can be seen from Figure 6 that the orange pareto front dominates the grey pareto front on nearly the entire objective space, here displayed for solution with a first objective value below $f_1(x) = 30.0$. Only for the tiny fraction of solutions with a flight time above 5500 days, this behavior breaks. Furthermore, from the close-up view displayed in Figure 7 it can be seen that the density of the orange pareto front is somewhat higher than the grey pareto front for the relevant solutions with a first objective function value below 4.0. This difference in the density indicates that those solutions are easier to achieve, which further adds to the decisiveness of sequence (3,3,3,5) in the MINLP scenario A.

V. Conclusions

A mixed-integer (MINLP) extension for the well-known continuous (NLP) Cassini1 benchmark problem was introduced. Comprehensive numerical results investigated the effort to solve several scenarios of the mixed-integer formulation and contrasted them to the complexity of the original continuous formulation. The three main findings are:

- The general MINLP formulation (denoted as scenario A in Section IV-A) of Cassini1 remains yet intractable and therefore appears to be even more difficult than the hardest continuous instance from the entire GTOP database, which is the Messenger (full version) benchmark. The difficulty of this particular MINLP instance appears to arise from a very decisive local optimum, which prevents the optimization algorithm from reaching the global optimal integer sequence (see discussion in Section IV-B).

- The MINLP formulation of Cassini1 becomes tractable by modifying the problem in such way that an additional artificial constraint excludes a single or few integer sequences from the search space. This appears to be a simple but effective strategy to solve the MINLP nonetheless.

- The varying MINLP instances (denoted as scenario B, C, D and E) exhibit a wide range of complexity. Compared to its continuous NLP counterparts, such MINLP formulations add a complexity that ranges from approximately just three times (scenario D, see Table VIII) to well above 130 times (scenario E, see Table VIII) on average.

Overall it can be concluded that the mixed-integer formulation of interplanetary space trajectory design problems remains a difficult optimization task and provides a challenge for years to come. Furthermore, the multi-objective mixed-integer formulation of such applications has been approached with preliminary numerical results, which offered a deeper insight into the reasons for the complexity regarding the mixed-integer aspect.

Acknowledgment

We would like to express our gratitude to the Advanced Concepts Team (ACT) for creating and publishing the GTOP benchmark set. Further we would like to thank the European Space Agency (ESA-ESTEC/Contract No. 21943/08/NL/ST) and EADS Astrium Ltd (Stevenage, UK) for their professional and financial support on the MIDACO development.

References

[1] Blum, C.: Ant colony optimization: Introduction and recent trends. Physics of Life Reviews Vol. 2, Issue 4, pp. 353-373, 2005.

[2] Ceriotti, M.: Global Optimisation of Multiple Gravity Assist Trajectories.
Ph.D. Thesis, Univ. of Glasgow, Glasgow, Scotland, U.K., 2010.

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

[4] Dorigo, M. Optimization, Learning and Natural Algorithms. PhD Thesis, Politecnico di Milano (Italy), 1992.

[5] Englander J., Conway B., Williams T.: Automated Interplanetary Trajectory Planning. Proc. AIAA/AAS Astrodynamics Specialist Conference, Guidance, Navigation, and Control, doi.org/10.2514/6.2012-4517, 13-16 August, Minneapolis, Minnesota, USA, 2012.

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

[7] Korosec P., Silc J., Bogdan Filipic B.: The differential ant-stigmergy algorithm. Information Sciences Vol 192, pp. 82-97, 2012.

[8] Napier S.W., McMahon J.W.: A novel multi-spacecraft interplanetary global trajectory optimization transcription. (Preprint) AAS 18-401, 2018.

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

[10] Schlueter M., Yam C.H., Watanabe T., Oyama A.: Parallelization Impact on Many-Objective Optimization for Space Trajectory Design. Int. J. Mach. Learn. Comput., 6(1), pp. 9 - 14, 2016.

[11] Schlueter, M., Gerdts, M.: The Oracle Penalty Method. J. Global Optim. 47(2), 293–325 (2010)

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

[13] Schlueter M., Munetomo M.: MIDACO parallelization scalability on 200 MINLP benchmarks. Journal of Artificial Intelligence and Soft Computing (De Gruyter), Vol. 7, Issue 3, pp. 171-181, 2017.

[14] Schlueter M., Wahib M., Munetomo M.: Numerical optimization of ESAs Messenger space mission benchmark. Proc. Evostar Conference (Springer), Amsterdam 19-21 April, pp. 725-737, 2017.

[15] Schlueter M., Munetomo M.: Multi-Objective Global Optimization for Interplanetary Space Trajectory Design. Proc. LeGO-2018 workshop, Leiden, The Netherlands, Sep 18-21, accepted, in press, DOI: 10.1063/1.5089988, 2018.

[16] Ceriotti M., Vasile M.: Automated multigravity assist trajectory planning with a modified ant colony algorithm. Journal of Aerospace Computing, Information and Communication 7(9), pp. 261-293, 2010.

[17] Ceriotti M., Vasile M.: MGA trajectory planning with an ACO-inspired algorithm. Acta Astronautica, 67(9-10), pp. 1202-1217, 2010.

[18] Vasile M., Martin J.M.R., Masi L., Minisci E., Epenoy R., Martinot V., Baig J.F.: Incremental planning of multi-gravity assist trajectories. Acta Astronautica, Vol. 115, pp 407-421, 2015.

[19] Izzo D., Sprague C., Tailor D.: Machine learning and evolutionary techniques in interplanetary trajectory design. arXiv preprint (arXiv:1802.00180), 2018.

[20] Izzo D., Maertens M., Pan B.: A Survey on Artificial Intelligence Trends in Spacecraft Guidance Dynamics and Control. arXiv preprint (arXiv:1812.02948), 2018.

[21] Simoes L.F., Izzo D., Haasdijk E., Eiben A.E.: Multi-rendezvous Spacecraft Trajectory Optimization with Beam P-ACO. Proc. EvoCOP 2017, LNCS 10197, pp. 141-156, 2017.

Appendix

Here the individual best known (and likely global) optimal solutions for the first four integer sequences are displayed. Each Table lists the individual objective function values $f_{1,...,4}(x)$, the constraint function values $g_{1,...,4}(x)$ and the continuous decision variable values $x_{1,...,4}$ in numerical high precision. Additionally, a *Bounds-Profil* is included for the decision variables, which illustrate the location of the individual decision variable regarding its lower and upper bound limits. Note that each solution differs significantly from each other.

TABLE X
SOLUTION FOR INTEGER SEQUENCE (3,2,3,5)

$f_1(x)$ =	**3.5007**28185527975	
$f_2(x)$ =	6187.206159009762814	
$f_3(x)$ =	-768.507162880275246	
$f_4(x)$ =	0.000694840431001	
$g_1(x)$ =	426.30012744	
$g_2(x)$ =	0.00863652	
$g_3(x)$ =	0.07658503	
$g_4(x)$ =	198483.82577383	
		Bounds-Profil
x_1 =	-768.507162880275246	—-x——————-
x_2 =	350.595125230118811	——————x–
x_3 =	234.191923967256514	———-x————-
x_4 =	55.791488954165523	-x———————-
x_5 =	1012.700812895418380	———x————
x_6 =	4533.926807962803650	——————x———

TABLE XI
SOLUTION FOR INTEGER SEQUENCE (3,3,3,5)

$f_1(x)$ =	**3.6307**81777451420	
$f_2(x)$ =	5975.069902670555166	
$f_3(x)$ =	-836.986643071973731	
$f_4(x)$ =	0.000786281657129	
$g_1(x)$ =	0.167E+10	
$g_2(x)$ =	0.119E+11	
$g_3(x)$ =	0.00005353	
$g_4(x)$ =	1025954.35092890	
		Bounds-Profil
x_1 =	-836.986643071973731	—x——————
x_2 =	207.913267357936888	—————x——
x_3 =	192.809629637231779	x——————
x_4 =	258.400175641384408	—————x———
x_5 =	910.784034067405287	———x————
x_6 =	4405.162795966596605	—————x——

TABLE XII

SOLUTION FOR INTEGER SEQUENCE (3,3,3,6)

$f_1(x)$ =	**4.7556**32436714976	
$f_2(x)$ =	8162.967666962844305	
$f_3(x)$ =	-825.243506159569279	
$f_4(x)$ =	0.000944397039300	
$g_1(x)$ =	0.204E+10	
$g_2(x)$ =	0.251E+11	
$g_3(x)$ =	0.00021720	
$g_4(x)$ =	143943.43791531	
		Bounds-Profil
x_1 =	-825.243506159569279	——x——————
x_2 =	171.311984098048015	x————————
x_3 =	235.345067209642110	———x—————
x_4 =	250.770721904264974	—————x———
x_5 =	1999.999846004226356	————————x
x_6 =	5505.540047746662822	———————x–

TABLE XIII

SOLUTION FOR INTEGER SEQUENCE (3,3,3,6)

$f_1(x)$ =	**4.9307**08003226642	
$f_2(x)$ =	6240.090241737491851	
$f_3(x)$ =	-789.766698392527246	
$f_4(x)$ =	2.754600375736785	
$g_1(x)$ =	0.00004152	
$g_2(x)$ =	2514.34670590	
$g_3(x)$ =	0.00006172	
$g_4(x)$ =	161293.47750763	
		Bounds-Profil
x_1 =	-789.766698392527246	—-x———————-
x_2 =	158.316489688202438	——-x——————-
x_3 =	449.385882385110790	———————-x-
x_4 =	54.709580747134851	-x————————-
x_5 =	1024.763444786727177	———x—————
x_6 =	4552.914844130316851	—————x———