Multi-Objective Global Optimization for Interplanetary Space Trajectory Design

Martin Schlueter[1,a)] and Masaharu Munetomo[1]

[1]*Information Initiative Center (IIC), Hokkaido University, Sapporo 060-0811, Japan*
schlueter@midaco-solver.com, munetomo@iic.hokudai.ac.jp

[a)]Corresponding author: schlueter@midaco-solver.com

Abstract. This contribution presents numerical results for global optimization of a multi-objective formulation of the well-known *Cassini1* interplanetary space trajectory benchmark published by the *European Space Agency* (ESA). The original *Cassini1* benchmark is a single-objective problem and frequently used as case study for global optimization algorithms due to its highly non-convex and very sensitive objective function. Here, the problem is extended to four objectives and thus classified as many-objective problem. The MIDACO optimization software represents an evolutionary hybrid algorithm and is used to solve the considered application in regard to two aspects. The first aspect considers the impact of massively parallelized co-evaluation in regard to reaching the global optimal solution and its influence on the solution objective space (particular the Pareto front shape). As a second aspect, the impact of a varying *BALANCE* parameter, which controls how much importance is given to each individual objective within a multi-objective preference scheme recently introduced as *Utopia-Nadir-Balance*, on the Pareto front shape is given. In regard to the first aspect, the results show that massive parallelization is an effective remedy to reduce the notoriously high number of sequential function evaluation while still maintaining a sufficient well distributed Pareto front. In regard to the second aspect, the results indicate that an exclusive focus on the first objective is preferable over a *BALANCE* parameter which distributes the preference over several objectives for this very special kind of application.

Keywords: Multi-Objective, Global Optimization, Massive Parallelization, Space Trajectory, MIDACO

INTRODUCTION

Interplanetary space mission trajectory design is 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 [3] 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 decision variables and four non-linear constraints (see [3] for details). This contribution considers a many-objective extension of this benchmark, which was introduced in Schlueter et al. [4] and which consists of four objectives. Table 1 list those four objectives together with their description, units and function properties. Note that the first objective is the original one. Ant Colony Optimization (ACO) in general and the MIDACO software in particular has been shown

TABLE 1. Four Objectives for Cassini1 Benchmark

Objective	Description	Unit	Function properties
F1	Total ΔV (including ΔV_∞)	Km/Sec	highly non-linear and non-convex
F2	Time of Flight	Days	linear (sum of variables $\sum_{i=2}^{6} x_i$)
F3	Launch Date	MJD2000	linear (first variable x_1)
F4	Launch ΔV_∞	Km/Sec	non-linear and non-convex

[1]In contrast to the easiest instance, the Messenger [7] benchmark is considered the most difficult instance of the GTOP benchmark set.

to be efficient for optimizing the design of interplanetary space trajectories, see for example [1], [2] or [5].

The first research aspect of this contribution concerns the impact of massively parallelized co-evaluation of solution candidates on the overall number of sequential steps (called *"Blocks"*, see Section 2.1 in [6] for details) required to solve the *Cassini1* benchmark to its best-known solution in regard to the first objective. The F1 objective describes the total ΔV (km/sec), which is equivalent to the required propulsion (fuel) of the mission. This objective is by far the most complex one considered among the four objectives listed in Table 1. Besides the expected reduction of sequential steps by applying massive parallelization, this contribution is also concerned with the impact of parallelization on the amount and distribution of non-dominated solutions among the four-dimensional objective space.

The second research aspect of this contribution concerns the impact of the recently introduced *Utopia-Nadir-Balance* [4] concept for multi-objective optimization. This concept represents a decomposition approach where a preference can be given to a certain objective or a combination of several individual objectives. The here presented results investigate if and how the set of non-dominated solution will change in regard to various *BALANCE* parameter values, which control the preference scheme.

NUMERICAL RESULTS AND CONCLUSION ON THE IMPACT OF MASSIVE PARALLELIZATION

This section presents numerical results of applying MIDACO 6.0 on the four-objective extended *Cassini1* benchmark. The parallelization factor **P**, which defines the amount of parallel processed solution candidates within one sequential algorithmic step, is varied from one to 1024 by exponentiating the value of two (see first column of Table 2). For each value of **P**, 30 individual test runs are conducted, each using a different random-seed and using the original lower bounds as starting point. An individual test run is considered successful and stopped, if the best known value (4.9307) in the first objective (F1) is reached within a precision of 0.1%[1] In regard to the many-objective nature of this problem, the first objective is set as exclusive target function to be minimized while the remaining three objectives are only filtered for non-dominance. Table 2 lists the best, worst and average results out of 30 test runs in regard to the required number of sequential steps (called *"Blocks"*) and the corresponding number of overall function evaluation. From Table 2 it can be seen that the average number of *Blocks* can be significantly reduced from **1,114,440** in the serial

TABLE 2. Numerical Results for 30 test runs with a varying parallelization factor **P** from 1 to 1024

	Best run out of 30		Worst run out of 30		Average over 30 runs		
P	*Blocks*	*Evaluation*	*Blocks*	*Evaluation*	*Blocks*	*Evaluation*	Speed Up Factor
1	255,017	255,017	2,976,114	2,976,114	**1,114,440**	1,114,440	1.00
2	175,815	351,630	3,569,693	7,139,386	**947,207**	1,894,414	1.18
4	120,180	480,720	2,348,015	9,392,060	**785,786**	3,143,145	1.42
8	45,035	360,280	443,219	3,545,752	**176,586**	1,412,695	6.31
16	68,058	1,088,928	420,124	6,721,984	**156,488**	2,503,819	7.12
32	106,382	3,404,224	212,239	6,791,648	**142,003**	4,544,125	7.85
64	4,409	282,176	189,996	12,159,744	**116,850**	7,478,434	9.54
128	1,868	239,104	129,772	16,610,816	**93,457**	11,962,525	11.93
256	3,922	1,004,032	117,153	29,991,168	**90,172**	23,084,202	12.36
512	3,300	1,689,600	115,993	59,388,416	**51,711**	26,476,424	21.55
1024	2,322	2,377,728	78,000	79,872,000	**26,654**	27,293,866	41.81

case (**P**=1) to **26,654** in the massively parallelized case of **P**=1024. In other words: While the MIDACO algorithm required about 1.1 million sequential function evaluation in the serial case to reach the best-known solution in high precision, the same solution could be reached within only about 26 thousand blocks of sequential function evaluation, whereas each such block contained 1024 individual function evaluation. Such reduction equals a speed up factor of about 41.81 times. The *"Evaluation"* column in Table 2 shows the number of total function evaluation corresponding to each **P**. In the serial case, the best run required 255,017 function evaluation in total to reach the best known solution.

From Table 2 it can be further seen that the number of *Blocks* for the best out of 30 runs shows a non-monotonic behavior and significant variance (e.g. 106.382 *Blocks* for **P**=32 versus only 4,409 *Blocks* for **P**=64). This great

[1]Note that 0.1% is the official required precision upon which ACT/ESA considers the benchmark to be solved.

variance is explained by the highly non-linear nature of the objective landscape, which implies a strong dependence on the random-seed used for each individual test run. A larger set of test runs will be necessary to reduce that effect.

The overall best test run was reached for a parallelization factor of **P**=128 and it required only 1,868 sequential steps. Figure 1 display the final set of non-dominated solutions respectively for the best run of the **P**=1 and **P**=1024 case. Note that while left side of Figure 1 shows visibly less non-dominated solutions than right side, it still captures the most relevant trade-off part of the front between the total ΔV (F1) and flight time (F2). The large difference in the algorithmic behavior between the serial and massively parallelized case is also well observable by the quite different scattering of the set of the last 30,000 evaluation, illustrated in the plots as tiny black crosses. In Figure 1 the position of the individual MIDACO solution among the Pareto front is highlighted as semi-transparent green hexagon. [1]

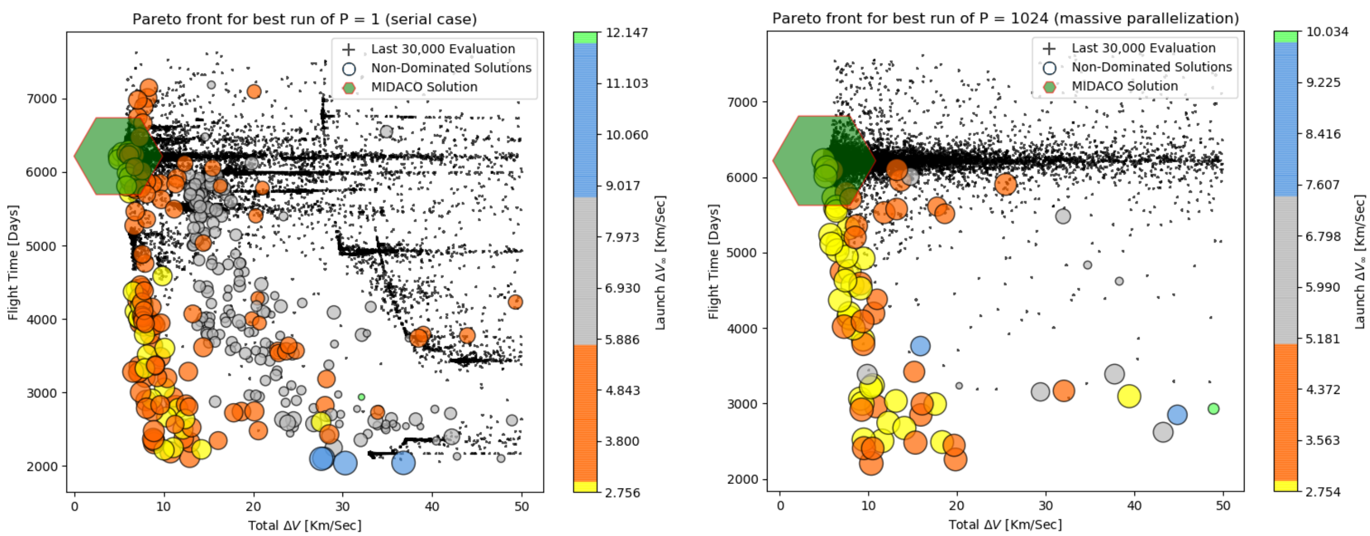

FIGURE 1. Pareto fronts for best run out of the serial case test runs (**P**=1) and the massive parallel case test runs (**P**=1024).

NUMERICAL RESULTS AND CONCLUSION ON THE *BALANCE* PARAMETER IMPACT

This section presents numerical results that investigate the impact of a varying *BALANCE* on the shape of the Pareto front as well as on the capability to reach the best-known solution in regard to the first objective. Table 3 reports the considered four different *BALANCE* parameters, their impact on the multi-objective search preference and the best result achieved in regard to the first objective function. Figure 2 illustrates the Pareto front for the case that the *BALANCE* is put exclusively on the first objective and the default case, where each of the four objectives is treated with equal importance. Figure 3 illustrates the Pareto front for two fine-tuning cases of the *BALANCE* parameter, the left side of 3 illustrates the results for a *BALANCE* parameter that onlye takes into account the first and second objective, while the right side of 3 illustrates the results for a *BALANCE* parameter that takes into account all four objectives but emphazises on the first objective. Note that in contrast to the results in the previous section (Figure 1) a finer epsilon tolerance for the Pareto dominance filtering was used to create the plots in Figure 2 and 3, which results generally in more displayed non-dominated solutions.

From Figure 2 and Figure 3 it can be seen that the *BALANCE* parameter that puts exclusive focus on the first objective delivers a Pareto front which is most detailed on interesting vertical trade-off edge between propulsion (Total ΔV) and flight of time (F2). Those results indicate that selecting the most challenging objective function (F1) as exclusive target is a valid strategy for this special kind of application.

[1]The third objective (launch date) is represented in the graphics via a varying marker symbol size. Smaller size means earlier launch date.

TABLE 3. Best value for first objective (F1) among the entire Pareto front for varying BALANCE parameters

BALANCE	Description of BALANCE impact on search effort	best F1 (Total ΔV) value
1.0000	putting the entire focus exclusively on the first objective F1	4.930804
0.0000	treating all four objectives with equal importance **(default)**	6.354540
0.9100	major focus on F1, minor focus on F2, zero focus on F3 and F4	6.895454
0.9111	major focus on F1, minor focus on F2, F3 and F4	6.669765

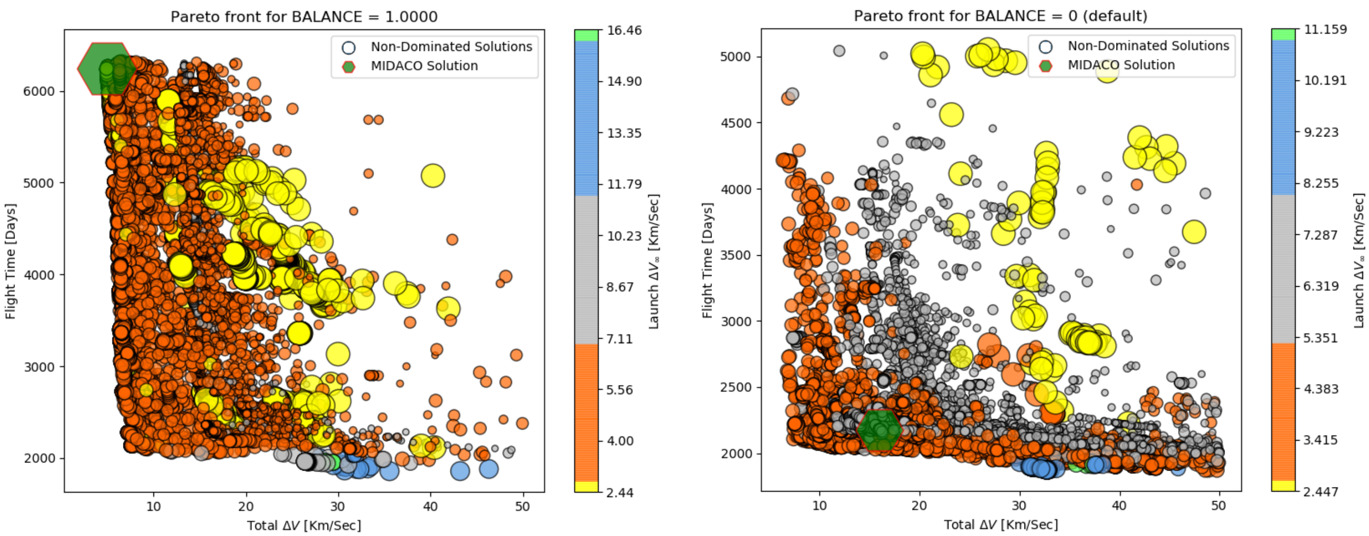

FIGURE 2. Pareto front results for a BALANCE parameter putting the importance entirely on the first objective (left) and a BALANCE parameter putting equal importance on all four objectives (right).

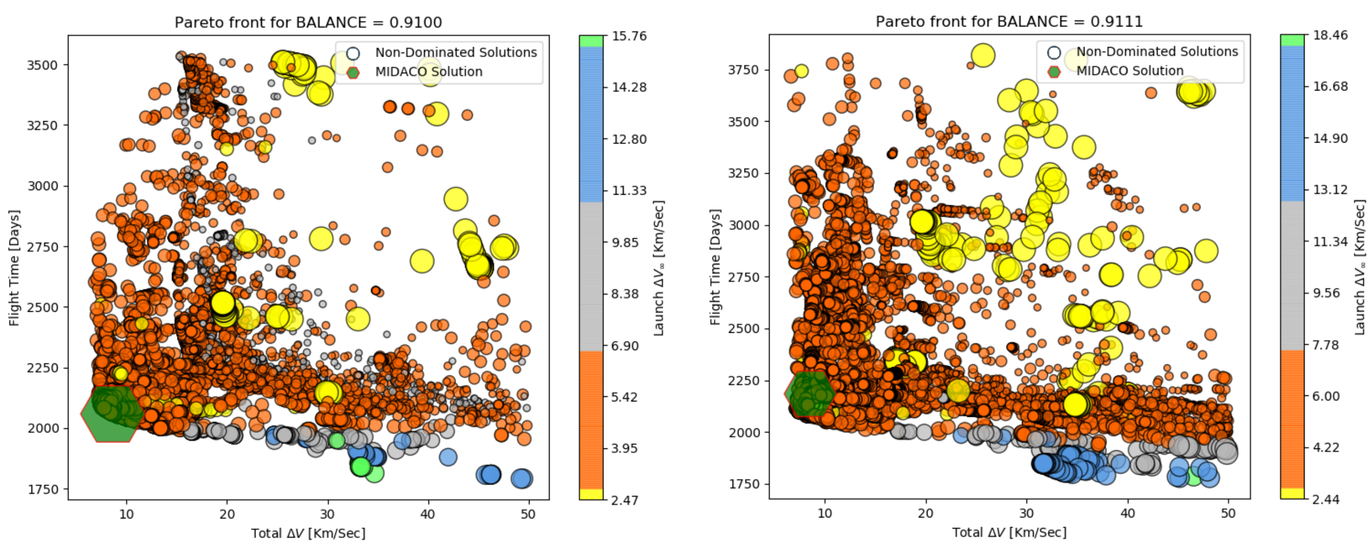

FIGURE 3. Pareto front results for a BALANCE parameter putting the major importance on the first objective and a minor importance only on the second objective (left) and a BALANCE parameter putting the major importance on the first objective and a minor importance on all remaining three objectives (right).

REFERENCES

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

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

[3] 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).

[4] 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).

[5] 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).

[6] 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).

[7] 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).