Solving Large MultiZenoTravel Benchmarks with Divide-and-Evolve

Alexandre Quemy, Marc Schoenauer, Vincent Vidal, Johann Dréo, Pierre
Savéant

Solving Large MultiZenoTravel Benchmarks
with Divide-and-Evolve

A. Quemy[1], M. Schoenauer[1], V. Vidal[2], J. Dréo[3], and P. Savéant[3]

[1] TAO Project, INRIA Saclay & LRI Paris-Sud University, Orsay, France
`alexandre.quemy@gmail.com, marc.schoenauer@inria.fr`,
[2] ONERA-DCSD, Toulouse, France
`Vincent.Vidal@onera.fr`
[3] THALES Research & Technology, Palaiseau, France
`{johann.dreo,pierre.saveant}@thalesgroup.com`

Abstract. A method to generate various size tunable benchmarks for multi-objective AI planning with a known Pareto Front has been recently proposed in order to provide a wide range of Pareto Front shapes and different magnitudes of difficulty. The performance of the Pareto-based multi-objective evolutionary planner DAE_{YAHSP} are evaluated on some large instances with singular Pareto Front shapes, and compared to those of the single-objective aggregation-based approach.

1 Introduction

Multi-Objectives Problems (MOP) involves several contradictory criteria to be optimized. The Pareto Set of a MOP is the set of the best trade-offs between these objectives, i.e., solutions that cannot be improved w.r.t. one objective without deteriorating at least another one. The projection of the Pareto Set on the objective space is called the Pareto Front.

Many benchmark suites exist for continuous multi-objective optimization, for which the exact Pareto Front can be analytically computed, and with known difficulties (e.g. dimensionality, shape of the Pareto Fronts, existence of local Pareto-optima, ...). For combinatorial optimization, the situation is not yet so clear, and whereas there exist famous benchmark problems of all sizes, their Pareto Fronts are generally not exactly known except the simplest ones (see e.g., MOCOLIB at `http://www.mcdmsociety.org/MCDMlib.html`).

The context of the present work is that of *AI planning*: a planning domain D is defined by i) a set of predicates, that define the state of the system when instantiated, and ii) a set of possible actions that can be triggered in states where their pre-conditions are satisfied, resulting in a new state. A planning problem instance $\mathcal{P}_D(I, G)$ is defined on a given planning domain D by a list of objects, used to instantiate the predicates to define the states, an initial state I and a goal state G. The aim is to come up with a *feasible plan*, i.e., a set of actions that, when applied in turn to the initial state, lead the system to the goal state, that is optimal w.r.t. a given measure: the number of actions, or the total cost of the plan when actions have non-uniform costs, or the total *makespan* (total
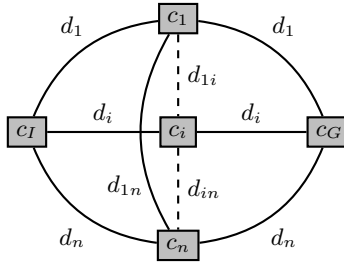
Fig. 1: A schematic view of a general MULTIZENOTRAVEL problem.

duration of the plan) when actions have durations, and can be run in parallel, as in the present work.

The present work presents the first results of $\text{DAE}_{\text{YAHSP}}$, an Evolutionary Pareto-based multi-objective planner [5] on large instances of MULTIZENO-TRAVEL domain with known Pareto Fronts, as proposed in [6]. The paper is organized as follows: Section 2 introduces the MULTIZENOTRAVEL benchmark suite, and the ZENOSOLVER algorithm that can derive the true Pareto Front for these instances. Sample very diverse experimental Pareto Fronts illustrate its versatility. Experimental results on some of the large MULTIZENOTRAVEL instances obtained by Divide-and-Evolve, the only Pareto-based evolutionary AI planner to-date [5], are compared with those of its single-objective version using the weighted sum aggregation on problems with non-convex fronts in Section 3.

2 MULTIZENOTRAVEL Benchmarks and ZENOSOLVER

MINIZENOTRAVEL is a simple temporal planning domain related to logistics, inspired by the well-known ZENOTRAVEL problem of IPC series[4]. It involves cities, passengers, and planes (see e.g., Figure 1); Planes can fly from one city to another when a link exists; Planes fly either empty, or carrying a unique passenger – and these are the only possible actions. In a MINIZENOTRAVEL instance (Figure 1), there are n central cities C_i, linked as a clique, and all are linked to the initial city C_I and the goal city C_G; the flight durations are d_i from city C_i to city C_I or C_G, and d_{ij} between cities C_i and C_j. There are t passengers and p planes, and all are in C_I in the initial state I, and all passengers must be in city C_G in the goal state G. The single objective version aims at minimizing the total makespan. Previous work [7, 5] proposed a multi-objective version of these benchmarks called MULTIZENOTRAVEL, by adding a *cost* c_i for landing in city C_i: the second objective is to minimize the *total cost* of the plan. More recent work [6] extended these benchmarks to problems of various complexity, and proved that such problems could provide Pareto Fronts of various shapes and difficulties, thanks to ZENOSOLVER, an exact Pareto solver.

---

[4] http://ipc.icaps-conference.org/

Table 1: Large instances: parameters and generation statistics.

Inst.	n	t	p	Generating functions		Pareto#	h	PPP(k)	Iter.(k)	Time
1	20	6	2			409	4015	1568220	3317140	16h46
2	3	21	2	$\frac{5}{2}i + \frac{(i \bmod 2)}{10}$	$\frac{5}{2}i + \frac{(i \bmod 2)}{10}$	61	861	53	233	2006s
3	10	10	3			383	7205	1056804	7918940	51h
4	8	26	25	$\sqrt{i}$	i	15	190	34176	60457	4240s

ZenoSolver is a C++11 software dedicated to generate and exactly solve MultiZenoTravel instances in cases where $d_i + d_j < d_{ij}$ for all (i,j). Firstly, it allows to tune the problem parameters in order to adjust the difficulty or to obtain different shapes of Pareto Fronts. In particular, vectors c and d are generated using two user-defined functions, f and g, such that $c_i = x_c f(i) + y_c$ and $d_i = x_d g(n-i) + y_d$, ensuring that both objectives are conflicting. ZenoSolver outputs the corresponding PDDL file (Planning Domain Definition Language, universally used to describe planning problems), that can be directly used by any standard AI planner, and computes the true Pareto Front (see all details in [6]).

We identified some large instances with very diverse front shapes and complexities, that could become a basic set of representative instances for MultiZenoTravel, allowing fair comparisons between various solvers and approaches. Table 1 gives the parameters used by ZenoSolver to build some of them, as well as some statistics about their complexity: The generating time strongly varies, from some minutes for Instance 1 up to 51h for Instance 3. The choice of the generating functions was purely empirical, guided by the fact that we wanted to obtain mainly piecewise concave fronts with uneven point distributions. This is why none of these fronts is linear, and most contain concave parts, i.e., parts where all points are above the segment made of the two extreme points. Unfortunately, this is not obvious on Figure 2, due to the large scale used here (but see [6] for some zooms). Note the small number of Pareto points of Instance 4, in spite of the complexity of this instance (26 persons), due to the small ratio $\frac{p}{t}$.

3 Multi-objective Experiments

3.1 Divide-and-Evolve

Based on the Divide-and-Conquer paradigm, this generic hybrid evolutionary approach has been originally introduced in [7]. The main idea to solve a planning problem $\mathcal{P}_D(I, G)$ is to find a sequence of states $S_1, \ldots, S_n$, and to use some embedded planner to solve in turn the series of planning problems $\mathcal{P}_D(S_k, S_{k+1})$, for $k \in [0, n]$ (with the convention that $S_0 = I$ and $S_{n+1} = G$). The generation and optimization of the sequence of states $(S_i)_{i \in [1,n]}$ is driven by an evolutionary algorithm, and each subproblems $\mathcal{P}_D(S_k, S_{k+1})$ is handled to an external 'embedded' planner. The concatenation of the corresponding plans (possibly with some compression step) is a solution of the initial problem. A more detailed presentation is given in [1].

3.2 Experimental Conditions

The MOEA used here is IBEA$_{H^-}$ [10], the Indicator Base Evolutionary Algorithm [10] using the Hypervolume Difference Indicator, that was demonstrated the best choice in previous work [5]. DAE$_{\text{YAHSP}}$ internal parameters have been tuned with PARAMILS [3], also using H^-. For each instance, 20 independent runs limited to 5400 seconds (1800 for instance 4) have been performed. This limit is arbitrary but early experiments on small MULTIZENOTRAVEL instances not shown here have empirically demonstrated (stagnation of the hypervolume for all runs) that indeed the algorithm had reached a stationary state within this limit. All performance assessments and comparisons have been done using the PISA platform [2].

3.3 DAE$_{\text{YAHSP}}$ on Large Instances

Attainment surfaces are displayed on Figure 2: the darker the region in objective space, the higher the probability to reach it (full white meaning that none of the 20 runs ever reached it). The attainment surfaces for the Instance 1 are uniformly distributed close to the true Pareto Front, even though very few Pareto optima were actually reached. The surfaces for Instances 2 and 3 are much further from the exact front (only 2 points are found for the Instance 3 out of 383). On the opposite, even if with a smaller budget, most of the actual Pareto optima are found for Instance 4, except on the most concave part.

We can notice that, even if n is higher for the Instance 4 than for the Instance 2, adding planes results here in Pareto front that is easier to reach. This is quite surprising since the search space for DAE$_{\text{YAHSP}}$ is increasing with p.

3.4 Pareto vs Weighted Sum Aggregation

Finally, let us have a quick look at some comparative results between the multi-objective version of DAE$_{\text{YAHSP}}$ and its single-objective version using a weighted sum of the objectives. The chosen instance is a concave instance with 30 cities (resulting in a Pareto Front made of 66 points) not displayed here. All experimental conditions are the same than in [4]. One aggregated run amounts to 11 independent runs, the weight α taking values from 0 to 1 by step of 0.1.

The attainment surfaces (Figure 3) show that in the case of Pareto approach, the exact Pareto Front is already delineated after 900s, even considering only the worst run. On the opposite, even the best of the 20 runs is still far from the Pareto Front apart from a few points that lie in the convex parts. This trend, though preliminary here, nevertheless confirms the well-known fact that weight sum aggregation has difficulties to reach the concave parts of Pareto fronts. However, using an archive shows that several non-dominated plans where found all over the Pareto front, strongly reducing the impact of the weight parameter. We hypothesize that this is due to the highly stochastic nature of YAHSP, that seems to be able to reach good results without the help of the genetic algorithm: A single individual can lead to several different objective vectors depending on
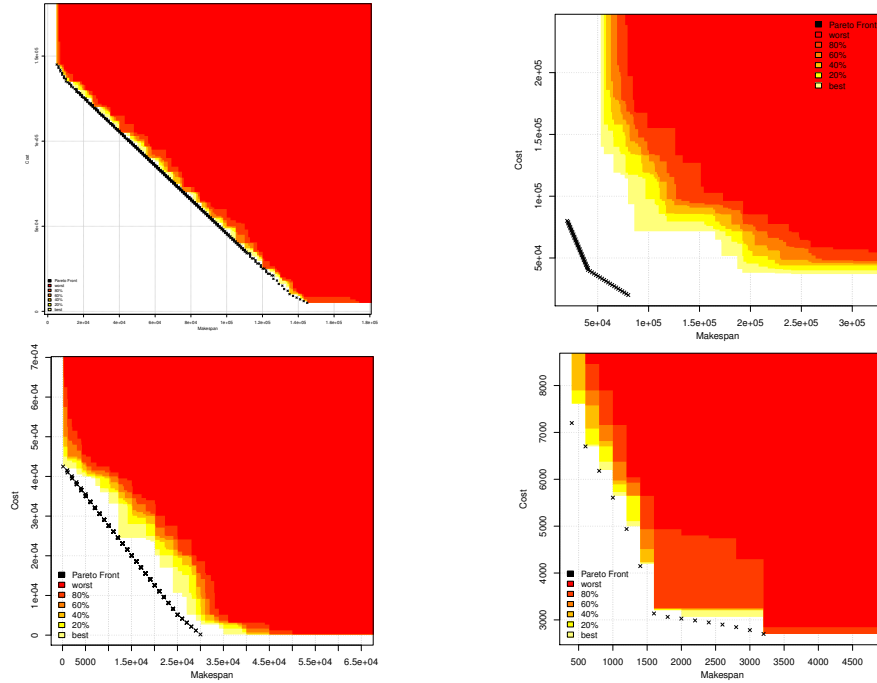
4

Fig. 2: Attainment surfaces for the Instances 1, 2, 3 and 4.

YAHSP strategy and random choices. The causality between the good structure of an individual and its fitness is thus very weak. Further work will study more deeply this hypothesis, and try to learn the relation between the individual structure and its ability to provide good plans.

4 Conclusion and perspectives

This paper has proposed some first experiments with the recently proposed MULTIZENOTRAVEL test suite for multi-objective AI planning [6], where the instance generator comes with ZENOSOLVER, an exact solver that is able to identify the true Pareto front for even very large instances. The complete code is publicly available at `https://descarwin.lri.fr`, making it easy for everyone to generate his/her own benchmark instances. However, we hope that the few typical instances that have been provided here, and that exhibit very different shapes of Pareto Fronts for very different levels of complexity, could be the starting point for a general benchmark for AI planning.

The results of $\mathrm{DAE_{YAHSP}}$ on some of these instances show the need for further improvement of the multi-objective search efficiency of MO-DAE. The results on the aggregation approach raise interesting issues regarding the respective usefulness of the evolutionary ($\mathrm{DAE_X}$) and the AI-planning (YAHSP) parts
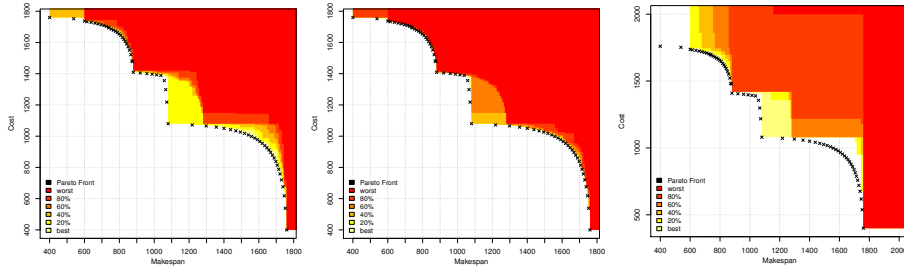
Fig. 3: Attainment surface for Pareto approach after 900s and 5400s (left, center) and for aggregation after 5400s (right).

of DAE$_{\text{YAHSP}}$. Further experiments are also needed, in which DAE$_{\text{X}}$ approach is used within other state-of-the-art decomposition algorithms (e.g., from the MOEA/D family [9], or using Tchebychev decomposition), and compared in detail to other non-Pareto multi-objective planners [8].

References

1. J. Bibaï, P. Savéant, M. Schoenauer, and V. Vidal. An Evolutionary Metaheuristic Based on State Decomposition for Domain-Independent Satisficing Planning. In R. Brafman et al., editor, 20th ICAPS, pages 18–25. AAAI Press, 2010.
2. S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler. PISA – A Platform and Programming Language Independent Interface for Search Algorithms. In *Evolutionary multi-criterion optimization*, pages 1–1. Springer, 2003.
3. F. Hutter, H.H. Hoos, K. Leyton-Brown, and Th. Stützle. ParamILS: An Automatic Algorithm Configuration Framework. *JAIR*, 36:267–306, 2009.
4. M.R. Khouadjia, M. Schoenauer, V. Vidal, J. Dréo, and P. Savéant. Multi-Objective AI Planning: Comparing Aggregation and Pareto Approaches. In M. Middendorf and C. Blum, editors, 13th EvoCOP, pages 202–213. LNCS 7832, Springer Verlag, 2013.
5. M.R. Khouadjia, M. Schoenauer, V. Vidal, J. Dréo, and P. Savéant. Pareto-Based Multiobjective AI Planning. In F. Rossi, editor, *Proc. IJCAI*. AAAI Press, 2013.
6. A. Quemy and M. Schoenauer. True Pareto Fronts for Multi-Objective AI Planning Instances. 2015. Submitted.
7. M. Schoenauer, P. Savéant, and V. Vidal. Divide-and-Evolve: a New Memetic Scheme for Domain-Independent Temporal Planning. In J. Gottlieb and G. Raidl, editors, 6th EvoCOP, pages 247–260. LNCS 3906, Springer Verlag, 2006.
8. M. Sroka and D. Long. Exploring Metric Sensitivity of Planners for Generation of Pareto Frontiers. In K. Kersting and M. Toussaint, editors, *6th STAIRS*, pages 306–317. IOS Press, 2012.
9. Q. Zhang and Hui Li. A Multi-objective Evolutionary Algorithm based on Decomposition. *IEEE Trans. on Evolutionary Computation*, 11(6):712–731, 2007.
10. E. Zitzler and S. Knzli. Indicator-based Selection in Multiobjective Search. In Xin Yao et al., editor, *PPSN VIII*, pages 832–842. LNCS 3242, Springer Verlag, 2004.