Visual Exploration of the Effect of Constraint Handling in Multiobjective Optimization

Tea Tušar[1,2(✉)], Aljoša Vodopija[1,2], and Bogdan Filipič[1,2]

[1] Jožef Stefan Institute, Ljubljana, Slovenia
[2] Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
{tea.tusar,aljosa.vodopija,bogdan.filipic}@ijs.si

Abstract. Constraint handling in multiobjective optimization is more complex than in single-objective optimization, where the values of the objective and constraints are easier to combine. To gain insight into the characteristics of constraint handling techniques (CHTs) for multiobjective optimization, we explore their effect independently from search methods. We regard CHTs as transformations that alter the problem landscape and visualize these modified landscapes. This helps us predict potential strengths and weaknesses for search methods. We then use a simple local search technique to test our predictions. Results of the experiments with six CHTs applied on 12 test problems show specific properties of the studied CHTs that can help us devise better CHTs in the future, as well as find suitable search methods for them.

Keywords: Constrained multiobjective optimization · Constraint handling technique · Problem landscape · Visualization

1 Introduction

Constraint handling in multiobjective optimization requires taking into account multiple (conflicting) objectives as well as constraints (often represented by the overall constraint violation). As such, it is more demanding than constraint handling in single-objective optimization, where the values of the sole objective and the overall constraint violation can be combined more naturally. Possibly for this reason, many constraint handling techniques (CHTs) in multiobjective optimization are closely intertwined with the search method [4,9,17,19], which makes it is hard to understand how much, when and why a particular CHT is more efficient than some other.

For example, as Ma and Wang show in [18], the efficiency of constrained multiobjective optimization algorithms heavily depends on the type of the problem. However, their study does not decouple CHTs from the optimization methods, meaning that its findings are tied to the frameworks of NSGA-II [5] and MOEA/D [27] that encompass the examined CHTs. Similarly holds for the work

M. Emmerich et al. (Eds.): EMO 2023, LNCS 13970, pp. 3–16, 2023.
https://doi.org/10.1007/978-3-031-27250-9_1

by Alsouly et al. [1] and our previous work [25], which connect problem landscape features with algorithm performance, but the considered algorithms differ in multiple mechanisms, not just in the CHT.

A notable exception in this regard is the study by Fukumoto and Oyama [11], which proposes a generic framework for incorporating CHTs into multiobjective optimization algorithms. It views a CHT separately from the search method and introduces a way to combine the two that covers dominance-based (e.g., NSGA-II [5]), decomposition-based (e.g., MOEA/D [27]), and indicator-based (e.g., IBEA [29]) multiobjective optimization algorithms. The experiments are then performed on different combinations of search methods and CHTs.

In this work, we explore the effect of CHTs independently from search methods, that is, as independently as possible. The goal is to enhance the understanding of their workings and provide intuition that can help guide the improvement of existing CHTs as well as find suitable search methods for particular CHTs. To this end, we regard CHTs as transformations that alter the problem landscape. We compute the CHT-based ranking of solutions from a grid approximation of the problem landscape to visualize it for various constrained multiobjective optimization problems (CMOPs). In this way, we are able to gain insight into the problem as 'seen' by an algorithm that uses a particular CHT. The CHT-based problem landscapes help us predict potential advantages and disadvantages for search methods. We then use a simple deterministic grid-traversing local search to test our predictions. The CMOPs used in this study are a combination of eight well known test CMOPs and four new, relatively simple problems with known properties that can help understand the characteristics of CHTs.

2 Background

2.1 Constrained Multiobjective Optimization Problems

We formulate a CMOP as follows:

$$\begin{aligned} \text{minimize} \quad & f(x) = (f_1(x), \dots, f_m(x)) \\ \text{subject to} \quad & g_i(x) \leq 0, \quad i = 1, \dots, p, \end{aligned} \tag{1}$$

where $x = (x_1, \dots, x_n) \in S$ is a search vector from the search space S, $f_i : S \to \mathbb{R}$ are objective functions and $g_i : S \to \mathbb{R}$ are inequality constraint functions. We do not explicitly include equality constraints as they can be formulated as inequality constraints with the help of a user-defined tolerance value.

The overall constraint violation of solution x is computed with

$$v(x) = \sum_{i=1}^{p} v_i(x), \tag{2}$$

where $v_i(x) = \max(0, g_i(x))$ is the constraint violation for constraint $g_i(x)$. Given that in this work we do not consider the constraints separately, we will be

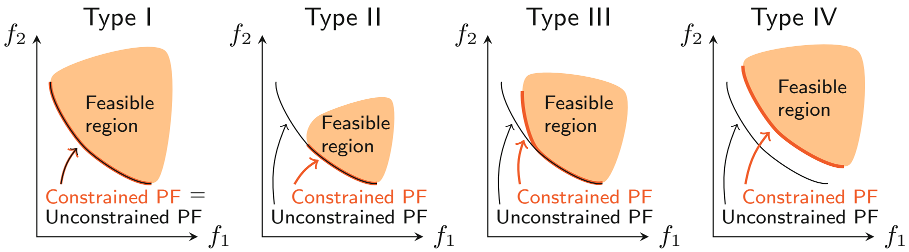

Fig. 1. The four types of CMOPs (adapted from [18]). The Pareto fronts (PFs) of the unconstrained/constrained problems are shown with thin black/thick orange lines. (Color figure online)

using the shorter term *constraint violation* instead of overall constraint violation to refer to $v(x)$ in the rest of this paper.

A solution x is *feasible* when it satisfies all constraints, that is, when $v(x) = 0$. The set of all feasible solutions is called the *feasible region*. A solution $x \in S$ *dominates* another solution $y \in S$ when $f_i(x) \leq f_i(y)$ for all $i = 1, \ldots, m$ and $f_j(x) < f_j(y)$ for at least one $j = 1, \ldots, m$. Additionally, a feasible solution $x^* \in S$ is *Pareto optimal* if there are no feasible solutions $x \in S$ that dominate x^*. All nondominated feasible solutions represent the *Pareto set*, and its image in the objective space is called the *Pareto front*.

When constraints are added to an otherwise unconstrained multiobjective optimization problem, this can affect the size and position of its Pareto set and front. The constraints that influence the Pareto set and front are called *active constraints*, while the remaining ones are termed *inactive constraints*. The degree of this change is the basis for the classification of CMOPs into types as proposed by Ma and Wang [18]. Figure 1 shows the four types, which range from no change to the Pareto front (Type I), to a reduced Pareto front (Type II), a partially displaced Pareto front (Type III), and finally an entirely different Pareto front (Type IV).

2.2 Constraint Handling Techniques

Our study comprises six methods for handling constraints in multiobjective optimization. In the following, we describe these CHTs and their known strengths and weaknesses.

One possible way of handling constraints (or rather, not handling them) is to simply ignore them and solve the problem as if it was an unconstrained one. We refer to this technique as *constraint violation ignored*. While such a strategy cannot be expected to yield good results on problems with active constraints and is therefore mostly omitted from comparison studies, it can be rather powerful for solving CMOPs where the constraints do not severely affect the optima, that is, problems of Type I (and, to some degree, Type II) [11].

Another method for handling constraints is to treat the constraint violation as an additional objective to be minimized[1]. We call this technique *constraint violation as objective*. One often mentioned drawback of this approach is that the additional objective can make the multiobjective optimization algorithm less efficient [17].

A very popular technique (due to being the default way of handling constraints in the algorithm NSGA-II) is the *constrained-domination principle* [5]. According to this principle, solution x is preferred to solution y if: (i) solution x is feasible and solutions y is infeasible, (ii) both solutions are feasible and x dominates y, or (iii) both solutions are infeasible and x has a lower constraint violation than y. The method is known to work rather well, except on problems with multimodal constraint functions [28].

The multiobjective version of the *epsilon-constraint method* [22] could be viewed as a relaxed variant of the constrained-domination principle, where solutions with the constraint violation lower than a predefined $\varepsilon \geq 0$ threshold are treated as feasible. More formally, the epsilon-constraint method prefers solution x to solution y when: (i) solution x dominates solution y and both have a small constraint violation ($v(x) \leq \varepsilon$ and $v(y) \leq \varepsilon$) or the same constraint violation, or (ii) solution x has a lower constraint violation than solution y. The optimization methods using the epsilon-constraint CHT usually gradually lower the value of ε during the algorithm run [2]. Choosing the appropriate starting value for ε as well as the mechanism to update it is nontrivial and problem-dependent.

Contrary to the methods that keep the objectives separate from the constraints, the *penalty function* transforms the objective values of infeasible solutions x to $f'_i(x)$ by either using the constraint violation (when there are no feasible solutions in the current population) or some penalty value that depends on the value of the objective, the constraint violation and the proportion r of feasible individuals in the current population [26]:

$$f'_i(x) = \begin{cases} v(x), & \text{if } r = 0 \\ (1-r)v(x) + r f_i(x) + \sqrt{f_i(x)^2 + v(x)^2}, & \text{if } r > 0 \end{cases}. \tag{3}$$

Suitably setting/adjusting the penalty value is recognized as a difficult task [17].

Finally, we also consider *stochastic ranking*, where the comparison of feasible solutions is done based on the dominance relation, while the infeasible solutions are compared either w.r.t. the constraint violation or the dominance relation—the decision between the two is done randomly [13].

3 Methodology

3.1 Test Problems

In order to explore the effect of CHTs, we need to select some test CMOPs. Because we aim to understand and visualize their landscapes, we choose problems with only two variables and two objectives. Ideally, the problems should

[1] The alternative variant, where each separate constraint violation is regarded as a new objective, is not considered in this work.

have various properties and be of different types [18]. We select eight problems from the existing well-known CMOP suites (C-DTLZ [15], DAS-CMOP [7], DC-DTLZ [16] and MW [18]) as well as create four new ones, CBB1–4, where CBB stands for Constrained BBOB [14] Biobjective (problem).

All four CBB problems were created by adding constraints to the first instance of the 2-D `bbob-biobj` problem F_1 (the double sphere problem) [3]. This is one of the easiest biobjective problems to solve as the Pareto set and front are linear and the problem landscape is unimodal (but not separable). However, when adding constraints to such a problem, it can become more difficult to solve while at the same time still easy to understand and interpret, which is why we created the CBB problems and added them to our test problem set.

The constraint function used in CBB1 is linear. It intersects the Pareto set of F_1 in such a way that the Pareto set of CBB1 consists of two connected linear parts. The constraint functions in the case of CBB2 and CBB3 are created by slightly shifting a single Gaussian peak function [12] with the same mean but a different covariance matrix, yielding in one case a problem of Type III (the Pareto set of CBB2 is formed by two linear parts of the original problem and one spherical that connects them) and in the other case a problem of Type IV (the entire Pareto set of F_1 is infeasible, the Pareto set of CBB3 consists of three disconnected spherical regions). Finally, CBB4 uses the inverted Gaussian peak function with three peaks as the constraint function (because the function is inverted, the peaks now form the feasible region). Again, the entire Pareto set of the original problem is infeasible, which yields a Type IV problem, whose Pareto set consists of two disconnected spherical regions. The exact definitions of constraints for problems CBB1–4 are provided in the supplementary material [24].

Thus we have 12 test problems in total, three of each type: Type I: DAS-CMOP3, DAS-CMOP5, MW14, Type II: C2-DTLZ2, DAS-CMOP1, DC1-DTLZ1, Type III: CBB1, CBB2, MW3, and Type IV: CBB3, CBB4 and MW11.

3.2 CMOP Landscape Visualization

First, we wish to visualize the problem landscapes of our 12 test problems (see Fig. 2). We can do so by approximating the search space with a grid of points. In this study, we always use a grid of 301×301 points[2]. We handle separately the feasible and infeasible regions of each problem. The feasible regions are visualized using the dominance rank ratio [3,10], which computes for each point on the grid the number of other grid points that dominate it and then visualizes them as a ratio of all grid points—using blue hues in the logarithmic scale to emphasize smaller values. The darker the color, the closer a point is to the Pareto set. Points with a domination rank of zero are Pareto optimal and visualized in black. The points in the infeasible regions are colored in red hues according

[2] Note that using a grid approximation inevitably results in some artifacts. For example, a linear Pareto set is in reality a line, but because of the approximation, some points adjacent to this line also result as nondominated, yielding a 'thick line'. The coarser the grid, the larger the artifacts.

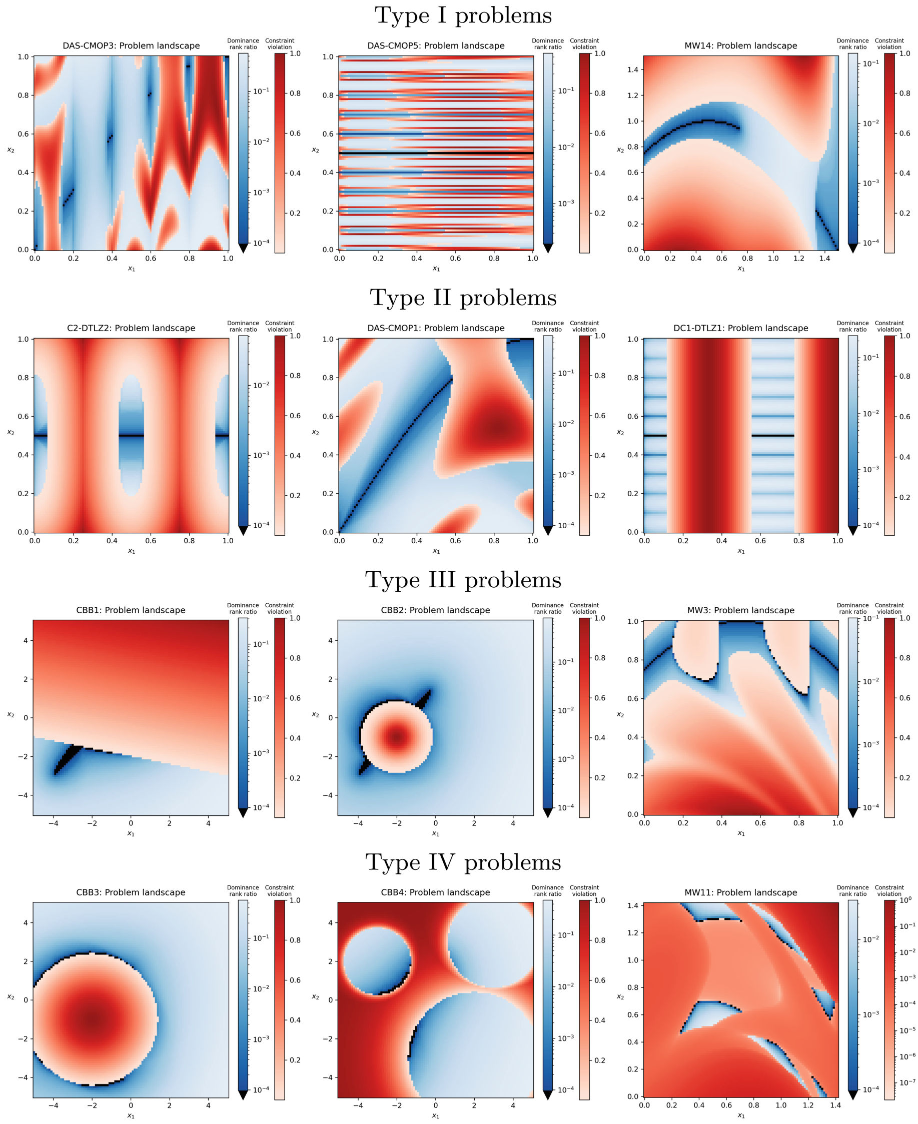

Fig. 2. Problem landscape plots for all 12 CMOPs used in this study. Each three problems of the same type are placed in the same row (from Type I at the top to Type IV at the bottom). Blue hues show the dominance rank ratio [3,10] in the feasible regions with black denoting the Pareto set. Red hues show the constraint violation in the infeasible regions. (Color figure online)

to their constraint violation values. Here, darker colors signify larger constraint violations. As we can see from Fig. 2, the chosen problems form a diverse selection of landscapes with various properties and Pareto set shapes.

Next, we want to see what becomes of the problem landscape when viewed from the perspective of a particular CHT. To this end, we compute for each CHT the rank that the CHT would assign to each point on the grid (compared to other points). Then, we visualize the ratio of this rank in blue hues (similarly as for the feasible problem regions of the original problem). Again, black is used to denote the grid points with the lowest rank—the optimal points according to the CHT. In this way, we gain a CHT-based problem landscape that assigns a single value to each grid point. We will show the visualizations of these landscapes in Sect. 4.2.

Note that if the Pareto set according to the CHT does not contain the entire Pareto set of the original problem, we can expect that an optimization algorithm using this CHT will have issues with convergence to the Pareto set.

3.3 Local Search

While already the visualization of a CHT-based landscape and comparison to the original problem landscape gives a good idea of some of the issues that a search method would encounter if it was used to find the optimum of such a problem, we wish to quantify these effects. Since any mechanism of a search method affects the behavior and interpretation of its results, we resort to a very simple, deterministic procedure—local search with a Moore neighborhood (each inner grid point in 2-D has eight neighbors).

Given a starting point on the grid, the local search iteratively moves to the best neighboring grid point that is not worse that the current point until a stopping criterion has been reached. The stopping criteria are: (i) the current point is optimal in the CHT-based landscape, (ii) the current point is better than all neighbors (it is a local optimum) (iii) all neighboring points have already been visited (to avoid cycling). In order to assure that this procedure is deterministic, the neighbors are always inspected in the same order (the north neighbor first then the rest in clockwise order) and an earlier neighbor always takes precedence over a later one when the ranks are tied among neighbors.

We can compute several quantities from a local search path on a CHT-based problem landscape. First, we can check (and visualize) if the final point of the path is Pareto optimal in the original landscape. If so, the path is denoted as successful (shown in orange) and the final point is visualized with a star. Otherwise, the path is deemed unsuccessful (shown in red) and the final point is denoted as a cross. In addition, simulating an optimization algorithm that chooses the best solution from its entire archive, we also record how many of the points on the path are Pareto optimal in the original landscape and how many are feasible. Of course, we also measure the path length (the number of points on the path).

4 Experiments

4.1 Experimental Setup

In our experiments, we apply the six CHTs from Sect. 2.2 to the 12 test problems from Sect. 3.1. We normalize both objectives and constraint violations to $[0, 1]$ before computing the CHT-based landscapes. There are no parameters to be set for the first three CHTs: constraint violation ignored, constraint violation as objective, and constraint-domination principle. We set the ε of the epsilon-constraint method to the 5th percentile of the constraint violation value of all infeasible grid points to mimic the initial parameter setting from [6]. Note that we do not vary the ε, therefore the epsilon-constraint method landscape in our study should be regarded as the landscape seen by the search method at the beginning of the optimization. Given that we do not use a population-based algorithm, we set the proportion r to the proportion of feasible points on the grid for the penalty function [26]. Finally, we use the recommended setting of 0.45 for the probability of comparing infeasible solutions according to the objective values in stochastic ranking [20].

We repeat local search 100 times, starting from 100 equally-spaced points on the grid for each combination of CHT and test problem. While the CHT-based landscapes are static for all steps of the local search for the first five CHTs, we use ten different stochastic ranking landscapes (in a loop) to mimic its stochastic behavior (each local search step uses one of the landscapes in turn).

4.2 Results and Discussion

CHT-Based Landscapes. We first inspect the CHT-based landscapes of the Type II problem CBB1, which is the easiest to understand (see the blue-hued landscapes in Fig. 3 and ignore the orange and red lines for now). When the constraint violation is ignored, the landscape obviously matches that of the original problem F_1, for which the Pareto set is linear. As approximately 2/3 of the apparent (as perceived by the CHT) Pareto set lie in the infeasible region, any search method that would ignore the constraint violation would spend a lot of effort in the infeasible region, making it inefficient.

When the constraint violation is treated as an objective, something interesting happens. The Pareto set of this CHT-based landscape contains not only the original Pareto set, but also a large region of otherwise infeasible solutions, which are nevertheless nondominated in the resulting 3-D objective space. While this does not happen on our problems of Type I and II, it appears on all six problems of Type III and IV. This would likely mislead a search method to regard a part of the infeasible region as optimal, which means that any optimization algorithm that uses this CHT needs to additionally check for feasibility of the apparent optimal solutions in order to be efficient.

Next, the landscapes of the constraint-domination principle, the epsilon-constraint method and the penalty function look very similar. However, note that the 'line' that we see in the landscape of the epsilon-constraint method

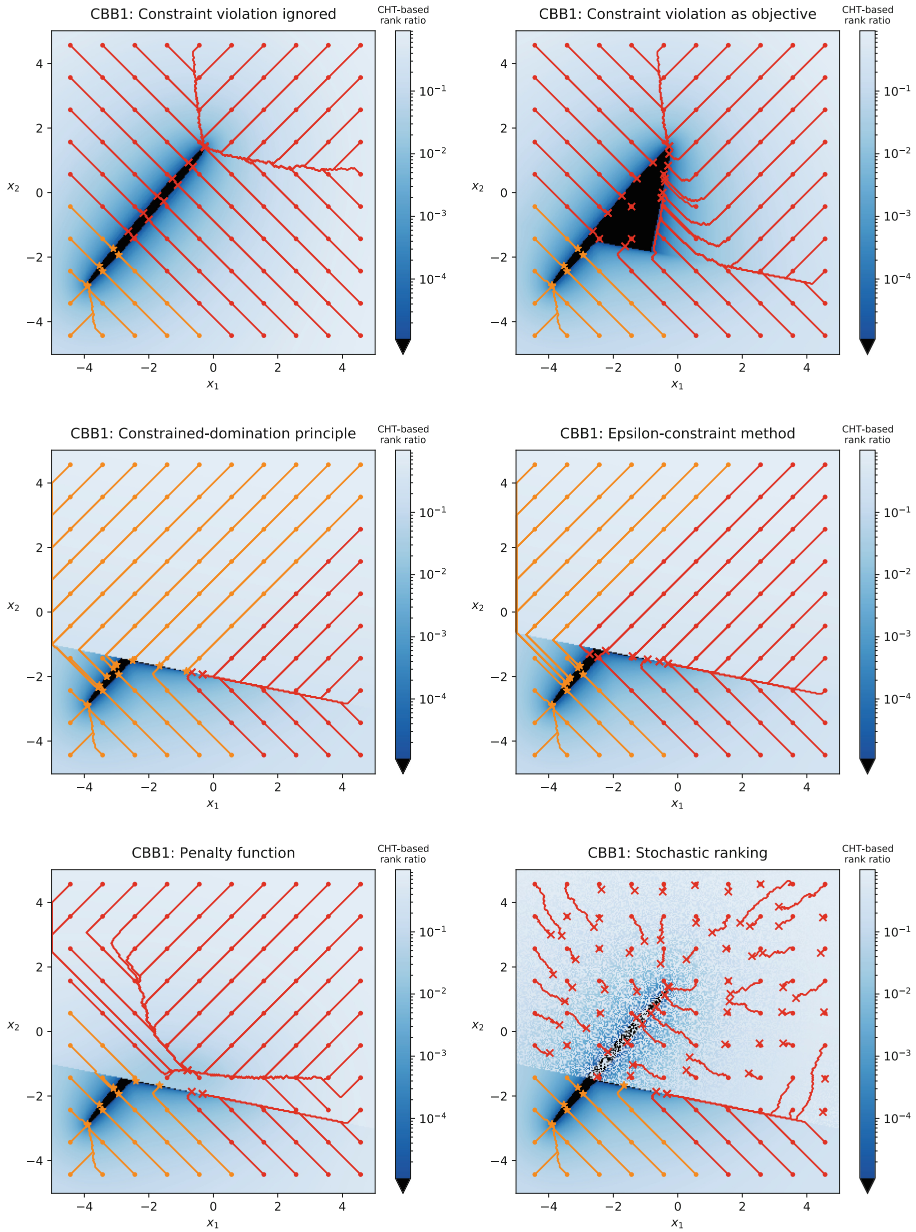

Fig. 3. Plots of the CHT-based landscapes for Type III problem CBB1 (in blue hues) for the six considered CHTs. Black denotes the Pareto set of these landscapes. Orange and red lines show the paths of local optimization starting in 100 different points shown with dots. If the path ends in a point that is optimal in the original problem landscape, the line is orange and it ends with a star, otherwise the line is red and it ends with a cross. (Color figure online)

does not match the feasible space boundary from Fig. 2 (the former is placed slightly higher than the latter). This means that for this CHT, the apparent Pareto set is misplaced, making the landscape misleading to the search method. This is why the optimization algorithms that employ this CHT need to gradually reduce the value of ε to 0 during the run, which then corresponds to the constraint-domination principle. Also, note that the penalty function-based landscape is also slightly different as the infeasible region of the original problem is darker close to the feasible space boundary. This adds some nonlinearity to the landscape with unclear influence on a search method.

The first landscape of stochastic ranking (of the ten used) clearly shows that the values of the infeasible region are randomly selected for each point separately between the original dominance rank and the constraint violation. This makes its landscape more rugged than the original one, which can pose problems to methods prone to get stuck in local optima.

Local Search Paths. If we now look at the local search (LS) paths in Fig. 3 (orange and red lines), we can confirm that these results are mostly in accordance with our predictions (LS with constrained violation ignored and constraint violation as objective is inefficient, LS with the epsilon-constraint method performs worse than with the constraint-domination principle and stochastic ranking is debilitating for local search in the infeasible region; we did not foresee the damaging effect of the penalty function CHT).

Similar reasoning about CHT-based landscapes and the corresponding local search paths could be applied also to the remaining problems. However, due to the lack of space we refer to the supplementary material [24] for these results.

Local Search Summary Results. The information summarizing the performance of local search paths can help us further analyze the CHTs. Figure 4 shows the number of optimal solutions vs. the proportion of feasible solutions for LS with each CHT on each problem. The number of optima is counted separately for the entire path (filled markers) and separately for just the ending path point (hollow markers). Note that these two quantities differ only for LS with constraint violation ignored and with the epsilon-constraint method, and only for Type III and IV problems. This happens because these two CHTs fail to guide local search on these problems, but still manage to cross the true Pareto set along the way.

Concentrating on the outcomes regarding the optimality of solutions (the y axis of plots in Fig. 4) we can immediately observe that the absolute worst results (regardless of the CHT) are achieved on DAS-CMOP5 and DC1-DTLZ1, which are multimodal and thus detrimental to local search. These two problems therefore do not help our analysis. Disregarding them, we can see the trend that the number of optimal solutions diminishes with increasing problem type, which could be expected. The relatively poor performance of local search with all CHTs except of the constrained-domination principle on the most basic problem

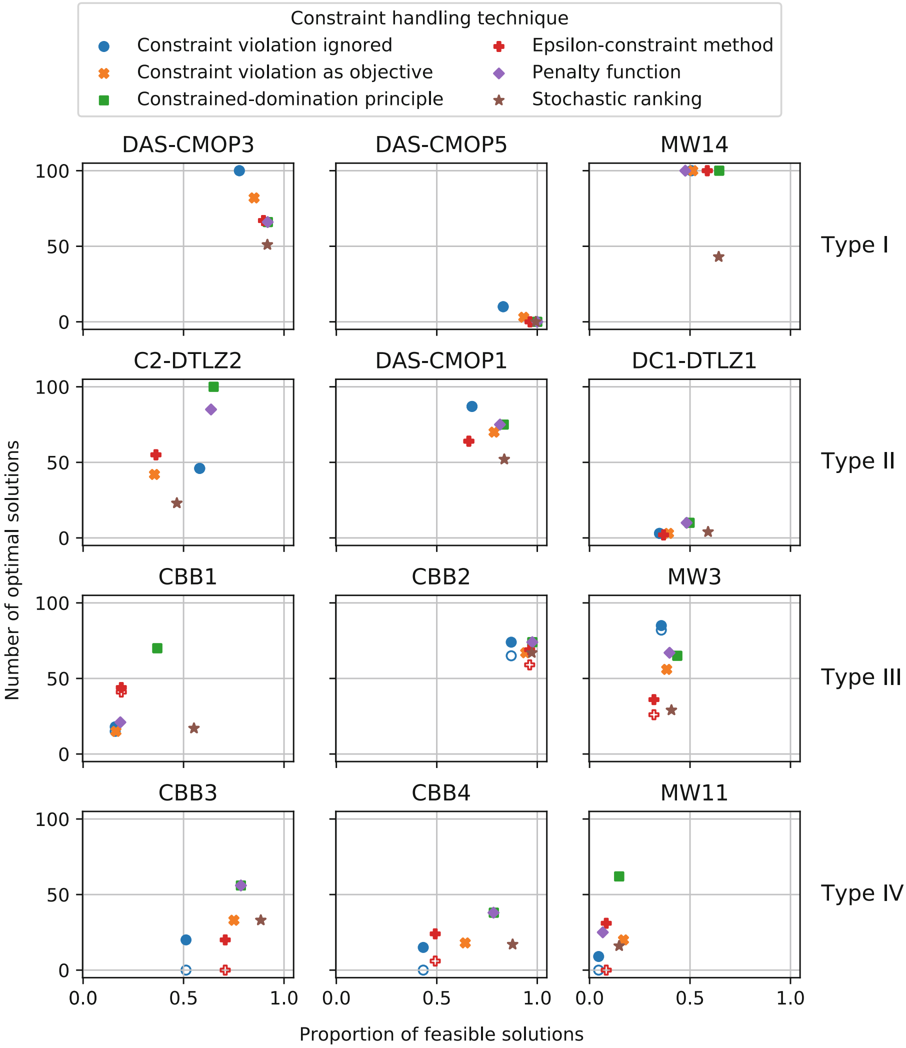

Fig. 4. The number of optimal solutions (y axis) vs. the proportion of feasible solutions (x axis) for local search with each CHT on each problem. Filled markers denote the number of all optimal solutions on the path, while the hollow markers show the number of final optimal solutions (the two differ only for constraint violation ignored and the epsilon-constraint method on problems of Type III and IV). (Color figure online)

CBB1 is quite disappointing. It shows that the constrained-domination principle is hard to beat and the other CHTs still have room for improvement.

We can further see that ignoring constraint violations is a very good strategy for solving problems of Type I, which confirms the results from [11]. Not so surprisingly, it is also one of the best CHTs for some Type II and III problems (those for which the intersection between the unconstrained and constrained Pareto set is large).

The performance of LS with constraint violation as objective never stands out (it is always in the middle). Similarly holds for LS with the epsilon-constrained method on problems of Type I, II and III (for Type IV, it shows a very poor performance). We also observe that the performances of LS with the constrained-domination principle and with the penalty function are mostly very similar with just a few exceptions. There (on C2-DTLZ2, CBB1 and MW11), the penalty function-based landscape is visibly different from the one by the constrained-domination principle, which has the undesired effect of guiding the local search away from the optima. These two CHTs are the only ones with a potential to solve Type IV problems with LS. Finally, the performance of LS with stochastic ranking is solidly among the worst.

If we look at the same results from the point of view of feasibility, we can see that, due to the 100 equally-spaced starting points of local search, there is generally not a large difference in the proportion of feasible solutions among the different CHTs. One (not so obvious) outlier here is LS with stochastic ranking, whose relatively good proportion of feasible solutions despite the otherwise poor performance stems from very short paths in the infeasible regions (the local search quickly becomes trapped in local optima of this very rugged landscape) rather than the CHT guiding the search towards the feasible region.

5 Conclusions

In this paper we proposed to look more closely at the various CHTs used for solving CMOPs in order to gain insight into their strenghts and weaknesses. This can help us devise better CHTs in the future, as well as find (more) appropriate search methods for particular CHTs. For example, we saw that constraint violation as objective requires additionally checking for feasibility, the epsilon-constraint method shifts the location of the apparent Pareto set and that the rugged landscape of stochastic ranking calls for a search method that can avoid being stuck in local optima. Our analysis has additionally confirmed findings from previous work [11,23] that problems of Type I (as well as some problems of Type II and III) are not helpful for benchmarking optimization algorithms on CMOPs as simply ignoring the constrains performs equally well.

As this work was limited to 2-D search and objective spaces we will consider generalizing our methodology to higher dimensions in the future. We would also like to similarly visualize the effects of dynamically changing CHTs and put more focus on local Pareto sets (possibly by using visualizations from [21] or [8]).

Acknowledgment. The authors acknowledge financial support from the Slovenian Research Agency (project no. N2-0254, young researcher program and research core funding no. P2-0209).

References

1. Alsouly, H., Kirley, M., Muñoz, M.A.: An instance space analysis of constrained multi-objective optimization problems (2022). https://arxiv.org/abs/2203.00868
2. Asafuddoula, M., Ray, T., Sarker, R.A., Alam, K.: An adaptive constraint handling approach embedded MOEA/D. In: IEEE Congress on Evolutionary Computation, CEC'12, pp. 1–8. IEEE (2012). https://doi.org/10.1109/CEC.2012.6252868
3. Brockhoff, D., Auger, A., Hansen, N., Tušar, T.: Using well-understood single-objective functions in multiobjective black-box optimization test suites. Evol. Comput. **30**(2), 165–193 (2022). https://doi.org/10.1162/evco_a_00298
4. Coello Coello, C.A.: Constraint-handling techniques used with evolutionary algorithms. In: Genetic and Evolutionary Computation Conference, GECCO'22, Companion Material, pp. 1310–1333. ACM (2022). https://doi.org/10.1145/3520304.3533640
5. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002). https://doi.org/10.1109/4235.996017
6. Fan, Z., Li, W., Cai, X., Huang, H., Fang, Y., You, Y., Mo, J., Wei, C., Goodman, E.: An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions. Soft. Comput. **23**(23), 12491–12510 (2019). https://doi.org/10.1007/s00500-019-03794-x
7. Fan, Z., et al.: Difficulty adjustable and scalable constrained multiobjective test problem toolkit. Evol. Comput. **28**(3), 339–378 (2019). https://doi.org/10.1162/evco_a_00259
8. Fieldsend, J.E., Chugh, T., Allmendinger, R., Miettinen, K.: A visualizable test problem generator for many-objective optimization. IEEE Trans. Evol. Comput. **26**(1), 1–11 (2022). https://doi.org/10.1109/TEVC.2021.3084119
9. Filipič, B., Vodopija, A.: Constraint handling in multiobjective optimization: tutorial. In: Presented at the IEEE World Congress on Computational Intelligence, WCCI'22. https://dis.ijs.si/filipic/wcci2022tutorial/
10. Fonseca, C.M.: Multiobjective genetic algorithms with application to control engineering problems. Ph.D. thesis, University of Sheffield (1995)
11. Fukumoto, H., Oyama, A.: A generic framework for incorporating constraint handling techniques into multi-objective evolutionary algorithms. In: Sim, K., Kaufmann, P. (eds.) EvoApplications 2018. LNCS, vol. 10784, pp. 634–649. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-77538-8_43
12. Gallagher, M., Yuan, B.: A general-purpose tunable landscape generator. IEEE Trans. Evol. Comput. **10**(5), 590–603 (2006). https://doi.org/10.1109/TEVC.2005.863628
13. Geng, H., Zhang, M., Huang, L., Wang, X.: Infeasible elitists and stochastic ranking selection in constrained evolutionary multi-objective optimization. In: Wang, T.-D., Li, X., Chen, S.-H., Wang, X., Abbass, H., Iba, H., Chen, G.-L., Yao, X. (eds.) SEAL 2006. LNCS, vol. 4247, pp. 336–344. Springer, Heidelberg (2006). https://doi.org/10.1007/11903697_43
14. Hansen, N., Auger, A., Ros, R., Finck, S., Pošík, P.: Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In: Genetic and Evolutionary Computation Conference, GECCO'10, Companion Material, pp. 1689–1696. ACM (2010). https://doi.org/10.1145/1830761.1830790
15. Jain, H., Deb, K.: An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, Part II: handling constraints

and extending to an adaptive approach. IEEE Trans. Evol. Comput. **18**(4), 602–622 (2014). https://doi.org/10.1109/TEVC.2013.2281534
16. Li, K., Chen, R., Fu, G., Yao, X.: Two-archive evolutionary algorithm for constrained multiobjective optimization. IEEE Trans. Evol. Comput. **23**(2), 303–315 (2019). https://doi.org/10.1109/TEVC.2018.2855411
17. Liang, J., et al.: A survey on evolutionary constrained multi-objective optimization. In: IEEE Transactions on Evolutionary Computation (2022). https://doi.org/10.1109/TEVC.2022.3155533
18. Ma, Z., Wang, Y.: Evolutionary constrained multiobjective optimization: test suite construction and performance comparisons. IEEE Trans. Evol. Comput. **23**(6), 972–986 (2019). https://doi.org/10.1109/TEVC.2019.2896967
19. Rahimi, I., Gandomi, A.H., Chen, F., Mezura-Montes, E.: A review on constraint handling techniques for population-based algorithms: from single-objective to multi-objective optimization (2022). https://arxiv.org/abs/2206.13802
20. Runarsson, T., Yao, X.: Stochastic ranking for constrained evolutionary optimization. IEEE Trans. Evol. Comput. **4**(3), 284–294 (2000). https://doi.org/10.1109/4235.873238
21. Schäpermeier, L., Grimme, C., Kerschke, P.: One PLOT to show them all: visualization of efficient sets in multi-objective landscapes. In: Bäck, T., et al. (eds.) PPSN 2020. LNCS, vol. 12270, pp. 154–167. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58115-2_11
22. Takahama, T., Sakai, S.: Constrained optimization by the ϵ constrained differential evolution with gradient-based mutation and feasible elites. In: IEEE Congress on Evolutionary Computation, CEC'06, pp. 1–8. IEEE (2006). https://doi.org/10.1109/CEC.2006.1688283
23. Tanabe, R., Oyama, A.: A note on constrained multi-objective optimization benchmark problems. In: IEEE Congress on Evolutionary Computation, CEC'17, pp. 1127–1134. IEEE (2017). https://doi.org/10.1109/CEC.2017.7969433
24. Tušar, T., Vodopija, A., Filipič, B.: Visual exploration of the effect of constraint handling in multiobjective optimization: supplementary material (2022). https://doi.org/10.5281/zenodo.7440416
25. Vodopija, A., Tušar, T., Filipič, B.: Characterization of constrained continuous multiobjective optimization problems: a feature space perspective. Inf. Sci. **607**, 244–262 (2022). https://doi.org/10.1016/j.ins.2022.05.106
26. Woldesenbet, Y.G., Yen, G.G., Tessema, B.G.: Constraint handling in multiobjective evolutionary optimization. IEEE Trans. Evol. Comput. **13**(3), 514–525 (2009). https://doi.org/10.1109/TEVC.2008.2009032
27. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007). https://doi.org/10.1109/TEVC.2007.892759
28. Zhu, Q., Zhang, Q., Lin, Q.: A constrained multiobjective evolutionary algorithm with detect-and-escape strategy. IEEE Trans. Evol. Comput. **24**(5), 938–947 (2020). https://doi.org/10.1109/TEVC.2020.2981949
29. Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: Yao, X., et al. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 832–842. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30217-9_84