Multi-objective mixed-integer evolutionary algorithms for building spatial design

Blom, K. van der

Author: Blom, K. van der
Title: Multi-objective mixed-integer evolutionary algorithms for building spatial design
Issue Date: 2019-12-11

Chapter 8

Towards General Multi-Objective Mixed-Integer Optimisation

So far this thesis has focused on the development of mixed-integer methods for multi-objective building spatial design. However, general methods for multi-objective mixed-integer (MOMI) optimisation are also needed, as stipulated by RQ5. This chapter takes first steps towards this goal.

Multi-objective optimisation for either only continuous or only integer variables is widely studied, the mixed-integer case is however largely neglected. In single-objective optimisation the mixed-integer case was successfully tackled by algorithms such as the Mixed-Integer Evolution Strategy (MIES) [69]. This chapter aims to extend the MIES algorithm for the multi-objective case.

It should be noted that other multi-objective mixed-integer approaches exist, such as the Enhanced Directed Search (EDS) method [67, 100]. Another technique is the direct zigzag method [99], but it was reported to be outperformed by the EDS method [100]. However, both the direct zigzag and the EDS methods do not distinguish between integer and nominal discrete (encoded by integers) variables like MIES does. Naturally, handling these variables separately may be advantageous.

Another approach to multi-objective mixed-integer optimisation is found in [105]. This work applies Bayesian Global Optimisation (BGO) to the problem. Although mixed-integer BGO does differentiate between integer and nominal discrete variables, BGO would require an extensive introduction and is therefore not included in this thesis. As such, this alternative is not elaborated on.

As in canonical evolution strategies (ES) [81, 91], one of the core principles of the MIES algorithm is automatic step size adaptation, i.e., the online adaptation of the strength of the stochastic perturbations. However, step size adaptation mechanisms for the single-objective case do not necessarily directly transfer to the multi-objective case. Furthermore, in [101] the authors analysed step size adaptation in evolutionary multi-objective optimisation for continuous problems, and reported the best performance when recombination is not used.

This chapter analyses mutation only approaches and step size adaptation in multi-objective mixed-integer evolution strategies. To this end, comparisons are made on both the performance in terms of diversity and convergence (combined in the hypervolume) to the Pareto front, as well as in terms of step size adaptation for the different variable types.

Section 8.1 introduces the considered algorithms and variations. Next, in Section 8.2 the experimental setup is introduced. Thereafter, the results are discussed in Section 8.3. Finally, Section 8.4 summarises the chapter and provides an overview of directions for future work.

8.1 Algorithms

Here three versions of a proposed algorithm will be introduced and compared. The core idea is the combination of the mixed-integer evolution strategy (MIES) [69] and the $\mathcal{S}$-metric selection evolutionary multi-objective algorithm (SMS-EMOA) [39]. Then, a variant is considered that does not use recombination, and one that selects offspring in a tournament during mutation. Each of the three variants is described in the following.

Firstly, an algorithm is considered that combines the canonical mixed-integer evolution strategy (MIES) as proposed in [69] with $\mathcal{S}$-metric selection and nondominated sorting as used in SMS-EMOA [39], as well as the $(\mu + 1)$ strategy considered there. In the multi-objective case recombination may have a disruptive effect on the step size adaptation mechanism. This is the result of different individuals navigating towards different parts of the Pareto front. However, this first variant will retain recombination for comparison purposes.

Secondly, an alternative is considered that only uses mutation, but is otherwise equivalent to the first algorithm. Instead of mutating the offspring resulting from recombination, mutation is applied to a uniformly at random selected individual.

Thirdly, an approach is considered that uses a tournament between mutants of the same parent as a local selection mechanism, but is otherwise equivalent to the second algorithm. In the tournament λ_k mutants are generated for the selected individual, rather than one. The mutant with the greatest hypervolume contribution is chosen as the winner, and enters $\mathcal{S}$-metric selection as usual. The idea is that competition between offspring may benefit step size adaptation. That is, the mutant with the better step size should also have the better performance, and should thus be selected.

In [69] no bounds were considered for continuous and integer step sizes. However, since these step size adaptation mechanisms were not designed with multi-objective optimisation in mind, step sizes might behave erratically and grow excessively. This could be one of the negative effects of recombining individuals that are navigating towards different parts of the Pareto front. To prevent this, step sizes for continuous and integer variables are given an upper bound equal to half the used variable range (as given in the next section). Step sizes of nominal discrete variables were already bounded in [69], and are bounded equivalently here.

8.2 Experimental Setup

To evaluate the algorithms three problems are considered: the multi-sphere (msphere) function in Equation 8.1, the multi-barrier (mbarrier) function in Equation 8.2, and the multi-objective optical filter (moptfilt) problem from [1, 5]. For the multi-objective case both the sphere and barrier functions can be adjusted with an offset for each term, such that continuous, integer, and nominal discrete optima are different in the second objective. The settings considered for each of these problems are shown in Table 8.1. Here $\mathbf{r}, \mathbf{z}, \mathbf{d}$ represent vectors of continuous, integer, and nominal discrete variables respectively, and n_r, n_i, n_d the dimensionality for each of them.

$$
\begin{aligned}
f_{sphere_1}(\mathbf{r}, \mathbf{z}, \mathbf{d}) &= \sum_{i=1}^{n_r} r_i^2 + \sum_{i=1}^{n_z} z_i^2 + \sum_{i=1}^{n_d} d_i^2 \rightarrow min \\
f_{sphere_2}(\mathbf{r}, \mathbf{z}, \mathbf{d}) &= \sum_{i=1}^{n_r} (r_i - 2)^2 + \sum_{i=1}^{n_z} (z_i - 2)^2 + \sum_{i=1}^{n_d} (d_i - 2)^2 \rightarrow min
\end{aligned}
\tag{8.1}
$$

$$\begin{aligned} f_{barrier_1}(\mathbf{r},\mathbf{z},\mathbf{d}) &= \sum_{i=1}^{n_r} \left(r_i^2 + \theta \sin(r_i)^2\right) \\ &\qquad + \sum_{i=1}^{n_z} A\left[z_i\right]^2 + \sum_{i=1}^{n_d} B_i\left[d_i\right]^2 \rightarrow min \\ f_{barrier_2}(\mathbf{r},\mathbf{z},\mathbf{d}) &= \sum_{i=1}^{n_r} \left((r_i - 2)^2 + \theta \sin(r_i - 2)^2\right) \\ &\qquad + \sum_{i=1}^{n_z} (A\left[z_i\right] - 2)^2 + \sum_{i=1}^{n_d} (B_i\left[d_i\right] - 2)^2 \rightarrow min \end{aligned} \tag{8.2}$$

problem	n_r	r range	n_z	z range	n_d	d range
f_{sphere}	5	$[0, 20]$	5	$[0, 20]$	5	$[0, 20]$
$f_{barrier}$	5	$[0, 20]$	5	$[0, 20]$	5	$[0, 20]$
$f_{optfilt}$	11	$[0, 1]$	N/A	N/A	11	$\{0, 1\}$

Table 8.1: Settings of the benchmark functions.

For the barrier function $\theta = 1$, and A is generated by Algorithm 6 from [69] with the parameter $C = 20$. This results in a vector of increasing integers that are occasionally swapped to create barriers for the optimisation algorithm, as shown in Equation 8.3. Further, $B_{i \in \{1,\ldots,n_d\}}$ is a set of n_d random permutations of the integer sequence $\{0, 1, \ldots, 20\}$, shown in Equation 8.4. Both A and B remain fixed for all experiments. Unlike in [69], here smooth wave-like barriers are used in the continuous part, rather than staircase-like barriers.

$$A = \begin{bmatrix} 0 & 1 & 2 & 4 & 6 & 3 & 5 & 7 & 8 & 9 & 11 & 12 & 10 & 14 & 15 & 16 & 13 & 17 & 19 & 20 & 18 \end{bmatrix} \tag{8.3}$$

A variant of the optical filter problem from [1, 5] with mixed variables and a second objective is considered here. Pairs of continuous and (binary) nominal discrete variables are used. When a binary variable is active, the corresponding continuous variable is used in the objective functions, otherwise it is ignored. If all bits are inactive a penalty of $(250, 1250)$ is returned.

$$B = \begin{bmatrix}
15 & 19 & 3 & 14 & 10 & 20 & 9 & 12 & 11 & 13 & 18 & 5 & 17 & 1 & 6 & 2 & 16 & 7 & 0 & 4 & 8 \\
14 & 11 & 9 & 20 & 16 & 15 & 0 & 10 & 2 & 13 & 3 & 4 & 1 & 5 & 17 & 6 & 7 & 12 & 8 & 18 & 19 \\
20 & 17 & 15 & 4 & 0 & 14 & 11 & 5 & 8 & 7 & 16 & 9 & 12 & 3 & 13 & 6 & 18 & 1 & 2 & 19 & 10 \\
14 & 5 & 18 & 6 & 9 & 11 & 8 & 2 & 20 & 7 & 12 & 13 & 3 & 0 & 10 & 15 & 16 & 4 & 1 & 17 & 19 \\
16 & 13 & 3 & 20 & 10 & 15 & 4 & 8 & 7 & 1 & 0 & 19 & 14 & 5 & 12 & 6 & 2 & 18 & 17 & 9 & 11 \\
19 & 4 & 11 & 17 & 16 & 12 & 0 & 7 & 6 & 18 & 8 & 1 & 5 & 14 & 10 & 15 & 2 & 3 & 9 & 20 & 13 \\
6 & 1 & 3 & 14 & 8 & 4 & 2 & 15 & 10 & 9 & 13 & 5 & 16 & 18 & 19 & 11 & 7 & 12 & 0 & 20 & 17 \\
16 & 1 & 17 & 15 & 0 & 12 & 11 & 18 & 13 & 7 & 19 & 14 & 2 & 8 & 9 & 3 & 10 & 20 & 4 & 6 & 5 \\
13 & 6 & 14 & 17 & 11 & 2 & 4 & 19 & 7 & 20 & 8 & 18 & 3 & 10 & 5 & 1 & 0 & 12 & 9 & 16 & 15 \\
18 & 20 & 15 & 4 & 11 & 9 & 0 & 6 & 5 & 8 & 12 & 17 & 19 & 3 & 14 & 16 & 10 & 7 & 1 & 13 & 2 \\
10 & 12 & 3 & 18 & 19 & 9 & 1 & 17 & 11 & 15 & 5 & 8 & 13 & 6 & 2 & 20 & 7 & 14 & 0 & 16 & 4 \\
6 & 19 & 0 & 14 & 1 & 7 & 17 & 12 & 16 & 18 & 11 & 4 & 13 & 8 & 15 & 3 & 20 & 9 & 10 & 2 & 5 \\
1 & 13 & 11 & 5 & 10 & 15 & 20 & 2 & 6 & 14 & 18 & 12 & 8 & 7 & 0 & 9 & 3 & 17 & 4 & 16 & 19 \\
18 & 7 & 13 & 3 & 6 & 8 & 20 & 11 & 2 & 15 & 12 & 9 & 4 & 19 & 0 & 5 & 1 & 14 & 10 & 17 & 16 \\
20 & 1 & 14 & 10 & 15 & 13 & 11 & 5 & 2 & 9 & 18 & 3 & 12 & 4 & 8 & 7 & 19 & 6 & 17 & 0 & 16 \\
5 & 6 & 9 & 19 & 14 & 3 & 12 & 17 & 13 & 11 & 15 & 10 & 0 & 2 & 4 & 20 & 18 & 16 & 1 & 8 & 7 \\
19 & 15 & 5 & 12 & 18 & 6 & 1 & 14 & 2 & 16 & 0 & 11 & 4 & 7 & 13 & 8 & 3 & 9 & 10 & 17 & 20 \\
3 & 13 & 17 & 9 & 6 & 12 & 4 & 20 & 14 & 18 & 5 & 10 & 2 & 8 & 19 & 11 & 1 & 7 & 0 & 15 & 16 \\
7 & 17 & 0 & 20 & 8 & 18 & 12 & 11 & 13 & 15 & 3 & 4 & 10 & 5 & 1 & 6 & 19 & 14 & 16 & 9 & 2 \\
4 & 1 & 3 & 15 & 19 & 8 & 16 & 14 & 10 & 6 & 18 & 5 & 0 & 7 & 20 & 17 & 11 & 9 & 13 & 12 & 2 \\
11 & 3 & 16 & 5 & 4 & 14 & 10 & 17 & 0 & 19 & 13 & 8 & 12 & 15 & 1 & 20 & 18 & 6 & 9 & 7 & 2
\end{bmatrix} \tag{8.4}$$

The original objective considers the transformation of a light wave by means of a filter that consists of layers of different materials from a limited set of materials (discrete variables). The layers can have different widths (continuous variables). The transformed waveform is compared to a target waveform and the root mean square error is measured, and to be minimised.In addition, a second objective is considered, the minimisation of the filter thickness:

$$f_{optfilt_2}(\mathbf{r}, \mathbf{d}) = \sum_{i=1}^{n_r} r_i d_i \rightarrow min. \tag{8.5}$$

8.2.1 Algorithm Settings

The canonical approach uses dominant recombination for the variables, and intermediate recombination for the step sizes as in [69]. All three approaches use single step size mode in all domains (continuous, integer, and nominal discrete), meaning a single step size per domain. Furthermore, $\mu = 10$, and a reference point $(2500, 2500)$ are considered for all approaches and objective functions. The tournament based approach uses a tournament of size 2. Step sizes are initialised to 25 % of the variable range for continuous and integer variables, and $\frac{1}{n_d}$ for nominal discrete variables. Step sizes are bounded to $[0, 10]$ for continuous, $[1, 10]$ for integer (upper bounds equal half of the range as mentioned before), and $[\frac{1}{n_d}, 0.5]$ for nominal discrete variables. An evaluation budget of 10 000 is used in order to be able to analyse the hypervolume and step size convergence during various phases in the optimisation process.

8.3 Results

For both the msphere and mbarrier problems the canonical MIES with $\mathcal{S}$-metric selection shows the fastest convergence in Figure 8.1, and outperforms the other approaches throughout the optimisation process. The mutation only approach has a slower start, but ultimately reaches only slightly worse hypervolume values. For both the msphere and mbarrier problems the tournament approach is clearly worse than the other two. Any possible advantages of the additional selection pressure in the tournament approach are clearly mitigated by the larger number of evaluations used per generation. On the moptfilt problem all three approaches quickly converge to a stable situation.

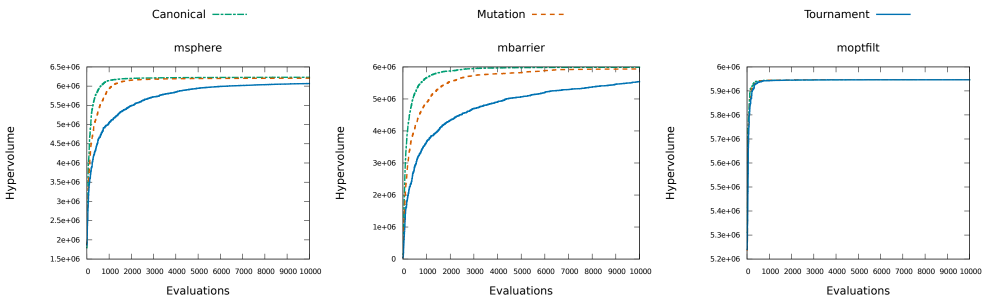

Figure 8.1: Mean hypervolume convergence over 25 repetitions.

Although it has more variables (22), it may be easier due to the smaller variable ranges that need to be searched.

The median attainment curves [49] in Figure 8.2 show that the canonical and mutation approaches find similar Pareto front approximations on the msphere and mbarrier functions, with the canonical approach remaining slightly better, as expected given the observed hypervolume convergence. All three approaches find very similar Pareto front approximations for the moptfilt problem, which suggests that they are close to the true Pareto front.

From Figure 8.3 it appears that step sizes σ for continuous and ς for integer variables stabilise reasonably well, whereas step sizes ζ for the nominal discrete variables show more erratic behaviour. However, Figure 8.4 shows that step sizes generated

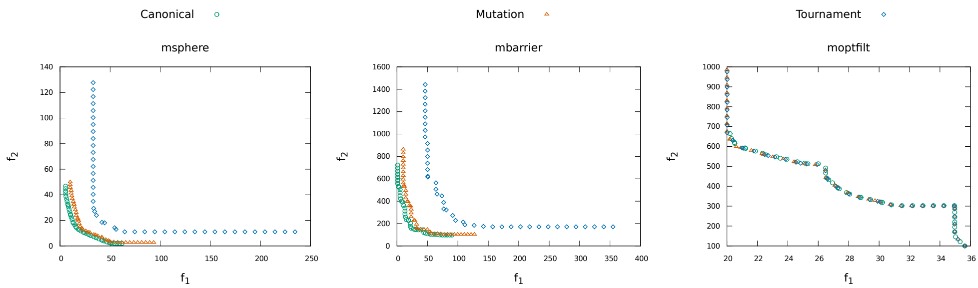

Figure 8.2: Median attainment curves over 25 repetitions.

for the offspring vary widely. The exception is the step size for continuous variables where the mutation only and tournament approaches do seem to stabilise. Although the tournament approach does so much later, this is likely due to its slower convergence. Thus, it appears only using mutation does indeed contribute to step size adaptation, but integer and nominal discrete step size adaptation have to be adjusted for the multi-objective case. Further, despite step sizes not adapting well when using recombination, it does result in better algorithm performance.

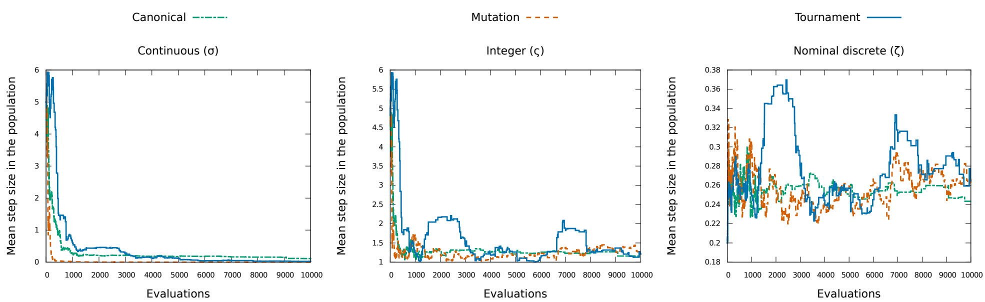

Figure 8.3: Mean step size in the population for each variable type, single run on the msphere problem.

8.4 Conclusion

8.4.1 Summary

In this chapter first steps have been taken towards general multi-objective mixed-integer optimisation, in accordance with RQ5. To this end the mixed-integer evolution strategy (MIES) [69] has been extended with multi-objective components from

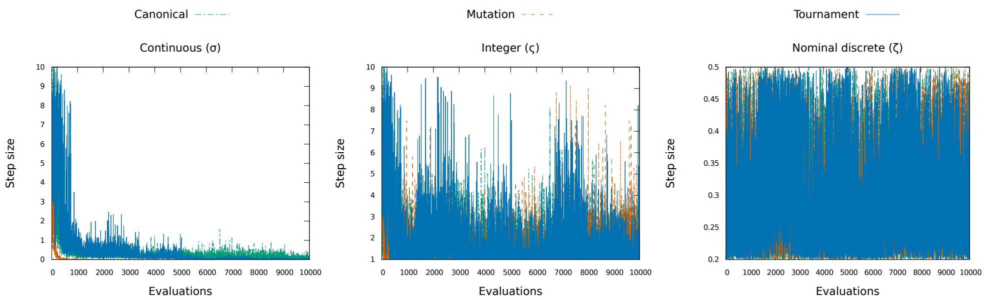

Figure 8.4: Step size per generated individual for each variable type, single run on the msphere problem.

SMS-EMOA [39]. Three variants of the considered algorithm have been introduced. Firstly, a variant that used MIES in its canonical form. Secondly, a variant that excluded recombination. Thirdly, a variant without recombination, but with a mutation tournament.

These approaches have been evaluated on three multi-objective mixed-integer problems, two test functions, and the real world problem of optical filter design. The two test functions considered all three types of variables: continuous, integer, and nominal discrete. In the optical filter problem only continuous and nominal discrete variables have been used.

The results, surprisingly,showed that the canonical variant of the algorithm performs best. Recombination seems to have a significant advantage in exploration, and is thus useful, despite its expected disruptive effect in step size adaptation for multi-objective problems. Further, the experiments showed that when only mutation is used, step sizes only adapt well for the continuous variables. Finally, mutation tournaments did not lead to sufficient additional selection pressure to offset the extra evaluation costs.

8.4.2 Future Work

Multiple directions of future work come to mind considering the experimental results. For instance, more work is needed to improve step size adaptation for the multi-objective case. First off, adaptation mechanisms have to be developed that allow integer and nominal discrete step sizes to stabilise. Second, step size adaptation that works together with recombination also has to be developed.

Another interesting question is what recombination is actually doing that makes it so effective. Is it, for instance, particularly useful for the navigation of the discrete landscape? Additionally, once the workings of recombination are better understood, it may be considered to switch partway through the optimisation process to a mutation-only approach. After all, if one is more effective in exploration, and the other in exploitation, this would be a beneficial strategy.

In addition to questions arising from the experiments presented here, the MOMIES algorithm has to be developed further in other directions as well. Throughout this thesis, the importance of constraint handling has become overtly apparent. As such, integrating constraint handling into the MOMIES algorithm is a natural next step to investigate.