

Point-Based Solution Approaches Based on the E-Constraint Method

In Section 4.1, we start by describing how we measure solution quality for problems with nonconvex constraints. Section 4.2 presents a 2-phase algorithm that efficiently distributes the points over the Pareto front with the goal to get the best approximation when only a limited number of points (portfolios) can be calculated. In Section 4.3 we discuss whether the problem formulation of the three problem types (problems with either a maximum cardinality, buy-in thresholds, or the 5-10-40-Constraint) has to be modified in order to allow the application of commercially available high performance mixed-integer solvers, and if so, how the modifications should look like. Section 4.4 presents heuristics for each nontrivial constraint type that, given a lower bound for the expected return, aim at calculating the portfolio with the smallest variance. These heuristics have been tested on 7 benchmark problems, and the obtained results are presented and analyzed in Section 4.5. Additionally, in this section, we report on tests that give an approximation on the effectiveness of the 2-phase procedure for point distribution. We conclude this chapter with a brief summary in Section 4.6.

4.1 Performance Measurement

Measuring performance in a multi-objective setting is difficult, because it requires the comparison of complete frontiers or approximations thereof and not only singular solutions/portfolios. A number of possible performance measures are discussed e.g. in Hansen and Jaszkiwicz [HJ98] or in Zitzler et al. [ZTL⁺03]. In the following, we will judge a generated front by its deviation from the *ideal front*, which is defined as the efficient frontier of the problem without non-convex constraints¹. This ideal front is an upper bound on the performance and can be computed efficiently with parametric quadratic programming, like e.g. the algorithm presented in Chapter 3. To measure the deviation, we calculate the area between the resulting front and the ideal front. One difficulty with area-based methods is to define the maximum variance and minimum return boundaries to calculate the area, see Figure 4.3. If these values are set far apart, extreme portfolios have a very large impact on solution quality. If they are set too close, some parts of the front may be cut off. Since the appropriate borders are not clear, we report on two values here: The area using the maximum variance and minimum yield portfolios from the ideal front (ideal delta-area), and the maximum variance and minimum yield from any asset in the available universe (max. delta-area).

Figure 4.3 illustrates how the success of an optimization run is measured for point-based approximations. If an approximation for the Pareto front can be described not only as a collection of points, but as a set of curves – as e.g. depicted in Figure 4.4 – the optimization results are measured analogously. The only difference now is that the area to be calculated does not have a stair-shaped appearance any more. Chapters 5 and 6

¹For the problem with 5-10-40-Constraint, the upper bound of all assets is set to 10%.

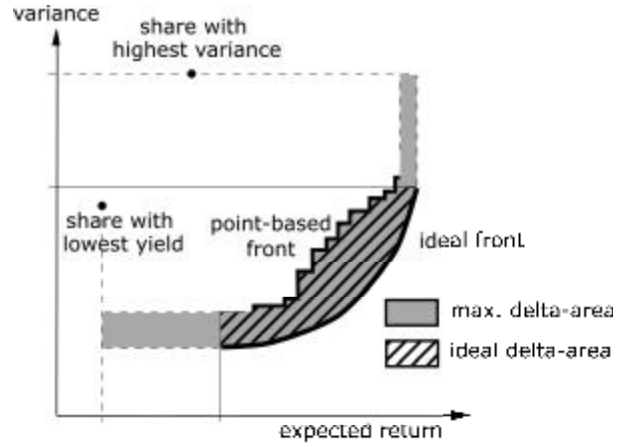


Figure 4.3: Ideal and maximum delta-area for point-based solution approaches.

present methods with this capability (envelope-based algorithms). An algorithm that reduces the negative effect of the “stairs” in Fig. 4.3 is presented in the following section.

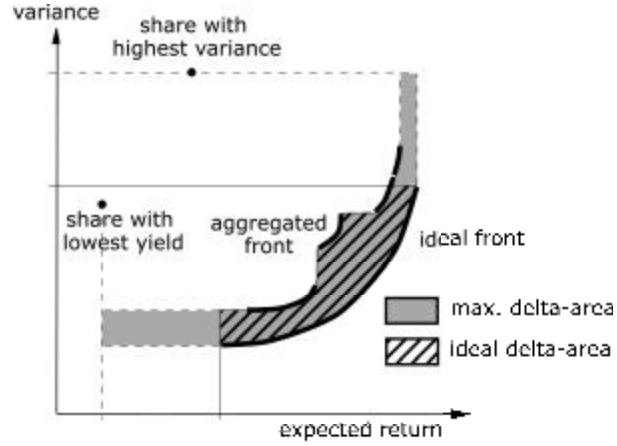


Figure 4.4: Ideal and maximum delta-area for envelope-based solution approaches.

4.2 Efficient Distribution of Points

When a limited number of points is used to approximate the Pareto front, there is always an approximation error due to the fact that nothing is known about the solutions between two adjacent points. This error exists even if all those points lie directly on the actual Pareto front.

5 An Envelope-Based MOEA

For this purpose, we use a multi-objective EA based also on the general framework of NSGA-II (see Deb [Deb01]). But instead of a solution being represented by a single point (portfolio) in the mean-variance space, now every solution is represented by an envelope in the mean-variance space. An exemplary population is depicted in Figure 5.2(a). The example shows that situations in which envelopes entirely dominate other envelopes occur very rarely. Instead, at many points, envelopes intersect with other envelopes. Even without intersections, envelopes may have dominated and nondominated parts. Thus, we had to adapt the nondominated sorting and crowding distance calculation to work with envelopes.

The basic idea can be described as follows:

We need to determine the *nondominated part of the set union of all envelopes* which will be called the (first) **aggregated front** in the remainder of this thesis. Following the idea of nondominated sorting, we assign Rank 1 to all individuals contributing at least partially to the first aggregated front. Then, we iteratively remove these individuals/envelopes from the population, and determine the aggregated front of the remaining individuals, assigning them the next higher rank, etc. The resulting ranking and the generated aggregated fronts are depicted in Figure 5.2(b).

It is clear that different individuals contribute differently to an aggregated front. Some may contribute only a small segment of the front, while others contribute large segments. Also, some parts of the aggregated front may be represented by several individuals. We use this information to rank the individuals within a front (substituting the crowding distance sorting in NSGA-II). For this purpose, we determine for each individual the length of the segment contributed to the aggregated front². Parts common to several individuals are shared among those individuals. For example, if the part contributed solely by an individual i has length 5, and a part with length 4 is shared with another individual j , the overall contribution of individual i is $5 + 4/2 = 7$. Within a front, individuals contributing more are considered more important. Parts of the efficient frontier not belonging to the aggregated front are not taken into account.

The actual implementation of the above envelope-based nondominated sorting is more involved than one would assume at a first glance. The main part – the algorithm that calculates the aggregated front – will be briefly described below. For more details, the reader is referred to Scheckenbach [Sch06].

5.4.1 Calculating the Aggregated Front

The basic principle to determine the aggregated front from a given set of envelopes works as follows:

The algorithm starts with the envelope that contains the corner portfolio with the highest overall yield, since this portfolio – and thus the envelope it belongs to – is not dominated by any other envelope. Moving from this point in the direction of decreasing yield, the algorithm iteratively selects those envelopes that for a given value of $E(\mathbf{x})$

²Although in principle it would be possible to calculate the true length of a segment, for reasons of simplicity we approximated the length by the Euclidean distance between the end points. Another possible criterion would have been the reduction in hypervolume if the individual is removed.

5 An Envelope-Based MOEA

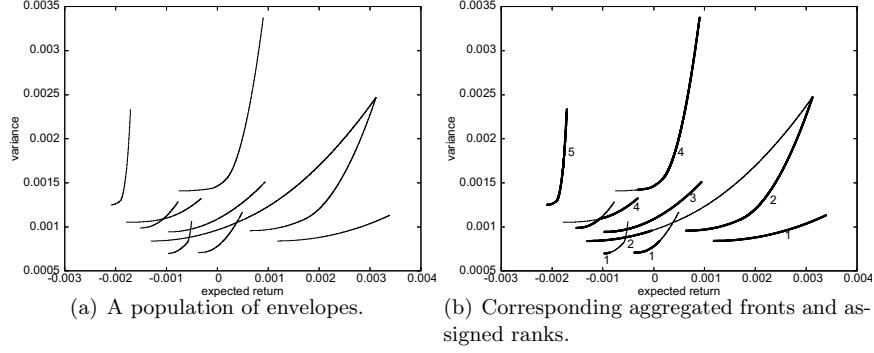


Figure 5.2: Ten randomly initialized envelopes and the five corresponding aggregated fronts for a problem with a maximum cardinality constraint of $k = 4$ for the Nikkei dataset (P5).

dominate all the other envelopes. The collection of these nondominated parts represents the aggregated front.

In order to simplify the algorithm description, we will first introduce several basic equations and definitions:

Each envelope is determined by a collection of corner portfolios. Every portfolio \mathbf{x}_α in the interval between two adjacent corner portfolios \mathbf{u} and \mathbf{v} is defined as a convex combination of these two corner portfolios:

$$\mathbf{x}_\alpha = \alpha \mathbf{u} + (1 - \alpha) \mathbf{v} = \mathbf{v} + \alpha \underbrace{(\mathbf{u} - \mathbf{v})}_{=: \mathbf{w}}, \quad \alpha \in [0, 1] \quad (5.5)$$

Its expected return and variance are:

$$E(\mathbf{x}_\alpha) = \alpha E(\mathbf{u}) + (1 - \alpha) E(\mathbf{v}) = E(\mathbf{v}) + \alpha E(\mathbf{w}), \quad \alpha \in [0, 1] \quad (5.6)$$

$$V(\mathbf{x}_\alpha) = (\mathbf{v} + \alpha \mathbf{w})^T \mathbf{Q} (\mathbf{v} + \alpha \mathbf{w}) = \mathbf{v}^T \mathbf{Q} \mathbf{v} + 2\alpha \mathbf{v}^T \mathbf{Q} \mathbf{w} + \alpha^2 \mathbf{w}^T \mathbf{Q} \mathbf{w}, \quad \alpha \in [0, 1] \quad (5.7)$$

By using Eq. 5.6 to replace α in Eq. 5.7, V can be expressed solely in terms of $E(\mathbf{x}_\alpha)$:

$$\begin{aligned} V(E_\alpha) &= \mathbf{v}^T \mathbf{Q} \mathbf{v} - 2\mathbf{v}^T \mathbf{Q} \mathbf{w} \frac{E_v}{E_w} + \mathbf{w}^T \mathbf{Q} \mathbf{w} \\ &\quad + 2 \left(\frac{\mathbf{v}^T \mathbf{Q} \mathbf{w}}{E_w} - \frac{\mathbf{w}^T \mathbf{Q} \mathbf{w}}{E_w^2} \right) E_\alpha + \frac{\mathbf{w}^T \mathbf{Q} \mathbf{w}}{E_w^2} E_\alpha^2, \quad E_\alpha \in [E_u, E_v] \end{aligned} \quad (5.8)$$

(For sake of brevity, $E(\mathbf{x}_\alpha)$, $E(\mathbf{v})$, and $E(\mathbf{w})$ have been replaced by E_α , E_v , and E_w . We have further assumed that $E_u < E_v$.)

5 An Envelope-Based MOEA

Eq. 5.8 demonstrates that in the mean-variance space, the image of each segment is part of a convex parabola, since

$$V''(E_\alpha) = \frac{\mathbf{w}^T \mathbf{Q} \mathbf{w}}{E_w^2} \geq 0$$

An envelope is therefore the continuous concatenation of parabola segments. When the aggregated front is considered, however, continuity usually can not be assumed, as e.g. Figure 5.2(b) shows.

In the further description of the procedure that calculates the aggregated front, the following expressions and abbreviations will be helpful:

The segment with the lowest expected return known to belong to the aggregated front will be called **current segment**, the envelope it is a part of will be referred to as the **current envelope**. The corner portfolio that marks the end of a segment and has the lowest expected return and variance is the **LCP**, the one with highest yield and variance is the **UCP**. The goal of each algorithm iteration is to find the next segment that is part of the aggregated front, the **succeeding segment**. The segment that was the current segment in the last iteration is referred to as the **preceding segment**.

There are three main scenarios how the transition from the current segment to the succeeding segment can take place:

1. The succeeding segment intersects the current segment.
2. The succeeding segment is reached by a vertical jump.
3. A horizontal or diagonal jump occurs at the end of the current segment.

The intersection scenario and the jump scenarios are treated separately:

Intersection

At the start, a list of candidates is compiled that comprises all segments that do not belong to the current envelope. Then, in the next step, segments are removed from the list if their exp. return coordinates don't overlap at all with exp. return coordinates of the current segment. We additionally used the convexity of the individual segments and of the whole envelope (cf. Markowitz [Mar87]) to further reduce the list size. For further details, the reader is referred to Scheckenbach [Sch06].

One possible way to calculate the intersection points is to equate the right hand sides of Eq. 5.8 for the current segment and the candidate segment.

Our procedure for the calculation is slightly different, however. Its main advantage is that it allows us to check easily for numerical errors when the succeeding segment has been chosen³ (see Scheckenbach [Sch06]).

In a first step, we calculate “virtual” corner portfolios that shorten the two segments to be considered in way that they completely overlap with respect to their expected return

³If the slope of two segments that intersect is nearly identical, the numerical error can – as a consequence of limited floating point precision – be quite large. There is no simple way to cope with this besides checking regularly and making corrections if necessary.

5 An Envelope-Based MOEA

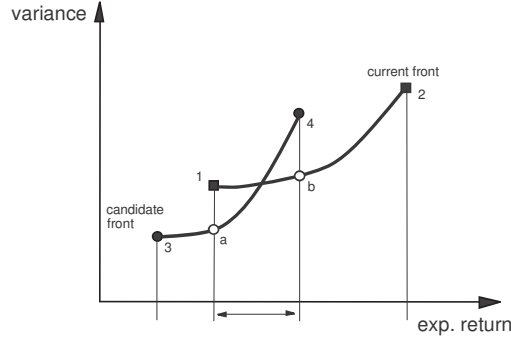


Figure 5.3: Creating virtual corner portfolios

coordinates. Suppose a situation as depicted in Figure 5.3, with the current segment defined by \mathbf{x}_1 and \mathbf{x}_2 , and the candidate segment determined by \mathbf{x}_3 and \mathbf{x}_4 . Then the virtual portfolios \mathbf{x}_a and \mathbf{x}_b are calculated as follows:

$$\mathbf{x}_a = \mathbf{x}_3 + \frac{E(\mathbf{x}_1) - E(\mathbf{x}_3)}{E(\mathbf{x}_4) - E(\mathbf{x}_3)}(\mathbf{x}_4 - \mathbf{x}_3)$$

$$\mathbf{x}_b = \mathbf{x}_1 + \frac{E(\mathbf{x}_4) - E(\mathbf{x}_1)}{E(\mathbf{x}_2) - E(\mathbf{x}_1)}(\mathbf{x}_2 - \mathbf{x}_1)$$

In the virtual segments ($\mathbf{x}_a \rightarrow \mathbf{x}_4$) and ($\mathbf{x}_1 \rightarrow \mathbf{x}_b$), the portfolios at an intersection can be described as in Eq. 5.5:

$$\mathbf{x}_{\text{candidate}} = \mathbf{x}_a + t_{\text{candidate}}(\mathbf{x}_4 - \mathbf{x}_a), \quad t_{\text{candidate}} \in [0, 1]$$

$$\mathbf{x}_{\text{current}} = \mathbf{x}_1 + t_{\text{current}}(\mathbf{x}_b - \mathbf{x}_1), \quad t_{\text{current}} \in [0, 1]$$

Since the intervals $[E(\mathbf{x}_a), E(\mathbf{x}_4)]$ and $[E(\mathbf{x}_1), E(\mathbf{x}_b)]$ are identical, $t_{\text{candidate}}$ and t_{current} are the same at an intersection point and can therefore be replaced by t . With this, equating the variances at the intersection point leads to:

$$\begin{aligned} \mathbf{x}_a^T \mathbf{Q} \mathbf{x}_a + 2(\mathbf{x}_4 - \mathbf{x}_a)^T \mathbf{Q} \mathbf{x}_a \cdot t + (\mathbf{x}_4 - \mathbf{x}_a)^T \mathbf{Q} (\mathbf{x}_4 - \mathbf{x}_a) \cdot t^2 = \\ \mathbf{x}_1^T \mathbf{Q} \mathbf{x}_1 + 2(\mathbf{x}_b - \mathbf{x}_1)^T \mathbf{Q} \mathbf{x}_1 \cdot t + (\mathbf{x}_b - \mathbf{x}_1)^T \mathbf{Q} (\mathbf{x}_b - \mathbf{x}_1) \cdot t^2 \\ \iff \\ \left[(\mathbf{x}_4 - \mathbf{x}_a)^T \mathbf{Q} (\mathbf{x}_4 - \mathbf{x}_a) - (\mathbf{x}_b - \mathbf{x}_1)^T \mathbf{Q} (\mathbf{x}_b - \mathbf{x}_1) \right] \cdot t^2 \\ + 2 \left[(\mathbf{x}_4 - \mathbf{x}_a)^T \mathbf{Q} \mathbf{x}_a - (\mathbf{x}_b - \mathbf{x}_1)^T \mathbf{Q} \mathbf{x}_1 \right] \cdot t + [\mathbf{x}_a^T \mathbf{Q} \mathbf{x}_a - \mathbf{x}_1^T \mathbf{Q} \mathbf{x}_1] = 0 \quad (5.9) \end{aligned}$$

5 An Envelope-Based MOEA

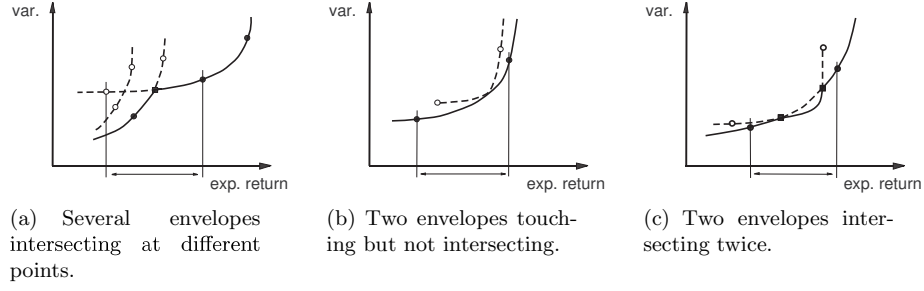


Figure 5.4: Different intersections constellations

This quadratic equation can have two solutions, or one, or none at all. If there is no solution, the parabolas of the two segments don't intersect. In the rare case of one solution, the candidate segment touches the current segment (see Fig. 5.4(b)) but remains dominated. When there are two solutions, i.e. there is a “real” intersection, a solution is only relevant if it is in $[0, 1]$, as only these result in an intersection in the interval $[E(\mathbf{x}_a), E(\mathbf{x}_b)]$. Usually, at most one of the two solutions will result in a feasible solution in the interval (Fig. 5.4(a)), but in rare cases, the two segments intersect twice in $[E(\mathbf{x}_a), E(\mathbf{x}_b)]$ (see Fig. 5.4(c)). Then the intersection point with the higher expected return has to be chosen.

When the complete list of candidates has been covered, the intersection point with the highest expected return in the interval $[E(\text{LCP}), E(\text{UCP})]$ of the current segment defines the succeeding envelope, but only if there is no vertical jump point with an even higher expected return.

Jump Points

A vertical jump to another envelope (Figure 5.5(a)) can occur everywhere on the current segment, but the succeeding segment has to be situated at the end of another envelope, namely the one with the highest expected return. It is therefore sufficient to check if the high-yield end of every envelope is in the interval $[E(\text{LCP}), E(\text{UCP})]$ of the current segment. Of all those envelopes fulfilling the condition, only those that start below the current segment dominate part of the current segment. Of these dominating envelopes, the one that starts with the highest expected return determines the succeeding envelope if no intersection with a higher expected return has been found.

A horizontal jump (Figure 5.5(b)) can only occur when the minimum variance portfolio (MVP) of an envelope is reached, i.e. when the current envelope ends and neither an intersection point nor a suitable vertical jump is found in the last segment. In this case all segments of the other envelopes are identified that have a UCP with higher variance and an LCP with lower variance than the variance of the MVP at the end of the current envelope. Of these candidate segments only those are viable to be the succeeding segment that have an LCP with a lower expected return than the MVP of the current envelope.

5 An Envelope-Based MOEA

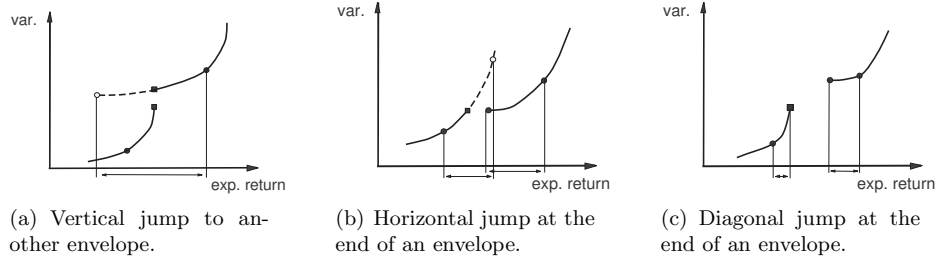


Figure 5.5: Possible jump discontinuities within the current mean interval

For each of the remaining candidates an artificial corner portfolio with the same variance as the MVP is constructed, and the one with the highest expected return is the candidate that is chosen as succeeding segment. Before the succeeding segment is definitely chosen, however, we have to check if a diagonal jump (Figure 5.5(c)) occurs that dominates the segment of the horizontal jump. For this to happen, there must be an envelope with a maximum exp. return smaller than that of the MVP and larger than the one of the artificial corner portfolio of the chosen candidate. Additionally, the variance of this envelope must be lower than the variance of the artificial corner portfolio. Is there more than one envelope that fulfills the conditions for a diagonal jump, the one that has the highest expected return is chosen.

When such a diagonal jump happens, it has to be recorded when the aggregated front is stored. Otherwise it can not be easily detected that there is a discontinuity in the front. (Horizontal and vertical jumps can be detected easily from the corner portfolios of the aggregated front, as there are two consecutive portfolios with either no change in variance or expected return.)

What was not covered above were cases that are the result of overlapping envelopes. For more details on how these are handled, the reader is referred to Scheckenbach [Sch06]. The whole procedure to assemble the aggregated front terminates when the portfolio with the lowest variance of all envelopes is reached.

5.5.2 Test Results

The results on all 4 benchmark problems are summarized in Tables 5.1 to 5.4. Table 5.1 reports on the max. delta-area, i.e. the area between obtained efficient frontier and ideal front with wide margins, for the cardinality constrained problem at the end of the run. The same information, but with respect to ideal delta-area, is provided in Table 5.2. As can be seen, E-MOEA significantly outperforms P-MOEA on all benchmark problems. In terms of the ideal delta-area, the relative performance of P-MOEA is somewhat better, indicating that it particularly has a problem in finding the portfolios with high expected return or low variance.

Table 5.1: Max. delta-area at the end of the run for E-MOEA and P-MOEA on test problems with cardinality constraint, average \pm std. error, all values in terms of 10^{-6} .

	P1 ($K = 4$)	P4 ($K = 4$)	P5 ($K = 8$)	P6 ($K = 8$)
P-MOEA	1.1613	2.7787	9.3292	4124.1363
standard error	± 0.0159	± 0.0521	± 0.2287	± 123.6716
E-MOEA	0.2275	0.8048	0.0561	5.9063
standard error	± 0	± 0.00003	± 0.0052	± 0.2939

The results for the problem with 5-10-40-Constraint look similar (see Tables 5.3 and 5.4), although the differences between P-MOEA and E-MOEA are generally smaller. Typical efficient frontiers for P1 and P5 with cardinality constraint are depicted in Figure 5.6. As can be seen, for the small problem (P1), both algorithms perform quite well. In fact, the figure zooms in on only a part on the front, as on a plot of the whole front, the differences would be hard to see. Still, E-MOEA clearly outperforms P-MOEA and is indistinguishable from the ideal front over large parts. For the larger

Table 5.2: Ideal delta-area at the end of the run for E-MOEA and P-MOEA on test problems with cardinality constraint, average \pm std. error, all values in terms of 10^{-6} .

	P1 ($K = 4$)	P4 ($K = 4$)	P5 ($K = 8$)	P6 ($K = 8$)
P-MOEA	1.0605	1.6981	2.1250	125.1611
standard error	± 0.0160	± 0.0175	± 0.0189	± 1.1580
E-MOEA	0.1371	0.5222	0.0123	2.4398
standard error	± 0	± 0.00003	± 0.0003	± 0.09637

Table 5.3: Max. delta-area at the end of the run for E-MOEA and P-MOEA on test problems with 5-10-40-Constraint, average \pm std. error, all values in terms of 10^{-6} .

	P1	P4	P5	P6
P-MOEA	1.3274	1.5107	1.7339	446.7284
standard error	± 0.0002	± 0.0197	± 0.0138	± 11.4963
E-MOEA	1.3019	1.2416	1.6511	344.7926
standard error	± 0.0007	± 0.0001	± 0.0047	± 3.8489

problem, P-MOEA does not seem to be able to come close to the performance of E-MOEA. In particular in the area of higher returns, its deficiencies are obvious. It seems that P-MOEA does not scale very well to larger problem sizes. (In Streichert et al. [SUZ04a, SUZ04b, SUZ03], the algorithm was only tested on small problems with up to 81 assets.) One reason may be that there are usually only few assets with a high return, and exactly those have to be combined in the portfolio to obtain an overall high return. Identifying the high-return assets out of a large set may prove difficult for the P-MOEA. E-MOEA on the other hand finds solutions hardly distinguishable from the ideal front also for the larger problems.

Table 5.4: Ideal delta-area at the end of the run for E-MOEA and P-MOEA on test problems with 5-10-40-Constraint, average \pm std. error, all values in terms of 10^{-6} .

	P1	P4	P5	P6
P-MOEA	0.2188	0.1028	0.0807	7.6607
standard error	± 0.00002	± 0.0009	± 0.0003	± 0.1116
E-MOEA	0.2023	0.0631	0.0700	4.9756
standard error	± 0.00001	± 0.0001	± 0.00006	± 0.0290

5 An Envelope-Based MOEA

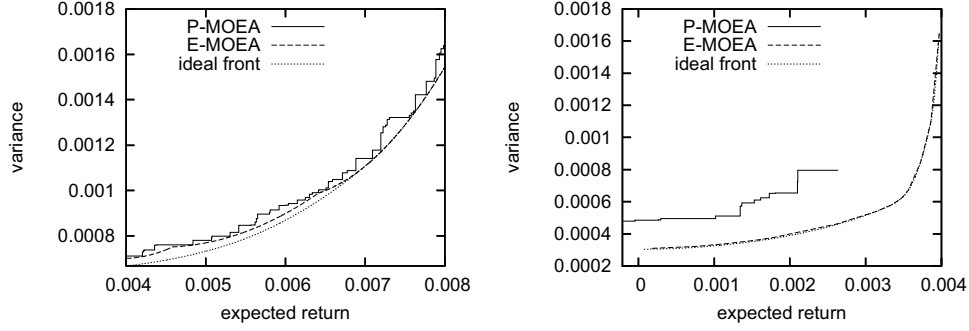


Figure 5.6: Typical fronts obtained on test problem P1 (left) and test problem P5 (right) with cardinality constraint.

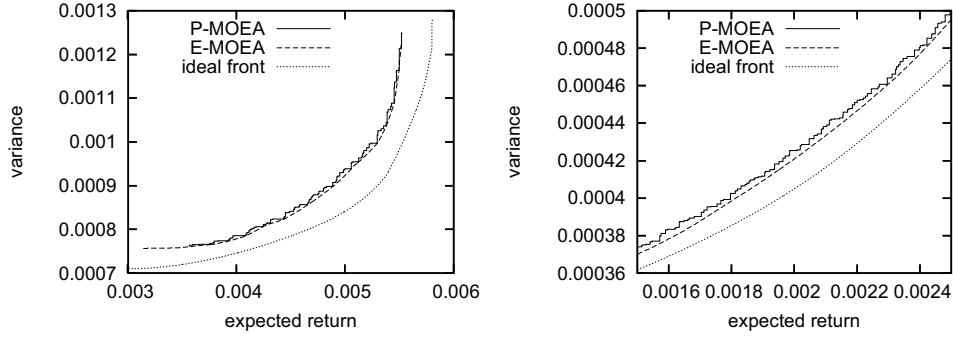


Figure 5.7: Typical fronts obtained on test problem P1 (left) and test problem P5 (right) with 5-10-40-Constraint.

For the 5-10-40-Constraint, the fronts obtained by P-MOEA and E-MOEA are much closer to each other, and further away from the ideal front. Still, the front obtained by E-MOEA dominates P-MOEA's front basically everywhere. Note that again for visibility, the plot for the larger problem only shows a segment of the overall front. When looking at the obtained solution quality in terms of max. delta-area over running time, it is clear that the advantage of E-MOEA over P-MOEA is significant throughout the run. Figure 5.8(a) looks at P5 with cardinality constraint. Clearly, E-MOEA starts out much better, and converges much faster than P-MOEA (for the small problem, E-MOEA even found the best solution within 5 out of the 250 available seconds in every single run). Note that we plot against running time. Because E-MOEA has to run a PQP algorithm for every individual, it can only evaluate about 13500 individuals during the 1000 seconds allowed, while P-MOEA generates and evaluates approximately 1,945,000 individuals in the same time frame.

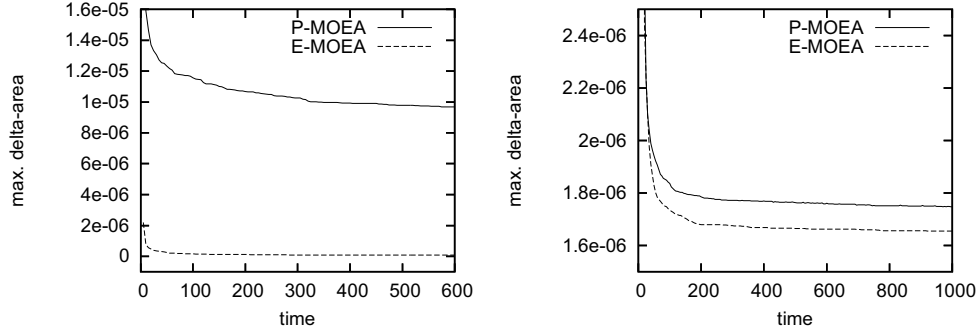


Figure 5.8: Convergence curves for P5 with cardinality constraint (left) and 5-10-40-Constraint (right).

For the problem P5 with 5-10-40-Constraint, E-MOEA also starts better than P-MOEA, then P-MOEA quickly catches up only to fall behind again (see Figure 5.8(b)). The difference in the number of individuals evaluated is even more striking here than for P5 with cardinality constraint, since the 5-10-40-Constraint does not allow to remove a large fraction of the assets for the PQP algorithm. While E-MOEA can generate only about 3000 individuals in the given 2000 seconds, P-MOEA generates approximately 3,475,000.

One explanation for the superiority of E-MOEA, besides being envelope-based, is certainly its in-built weight optimization by the PQP algorithm. This effect is visible in Figure 5.9, which depicts the solution quality of randomly generated solutions for both the P-MOEA and the E-MOEA. Clearly, E-MOEA has a much better start, as the majority of randomly generated envelopes is clearly better than the majority of randomly generated portfolios.

5.6 Concluding Remarks

Parametric quadratic programming algorithms – like the one described in detail in Chapter 3 – are a very efficient way to calculate the whole efficient frontier for a standard mean-variance portfolio selection problem. The inclusion of nonconvex constraints such as a maximum cardinality constraint, buy-in thresholds, or the 5-10-40 rule from the German investment law, however, renders the feasible region nonconvex and thus prevents us from applying this fast and efficient method. In the previous chapter we have therefore – similar to other researchers – resorted to solving these problems point-based. We approximated the efficient frontier by iteratively solving a sub-problem with a fixed expected return E_f , and repeated this for many different settings of E_f . Most other publications work within a meta-heuristic framework, but nevertheless, the approaches are all point-based.

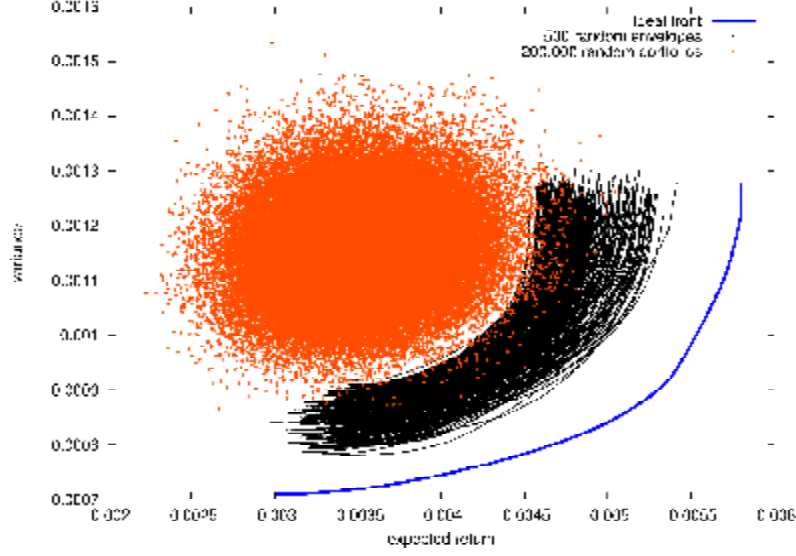


Figure 5.9: Randomly initialized populations of envelopes and portfolios for the 5-10-40 problem and the Hang Seng Dataset (P1).

In this chapter, we have proposed a new envelope-based multi-objective evolutionary algorithm (E-MOEA), which is a combination of a multi-objective algorithm with an embedded solver for parametric quadratic programming. The task of the MOEA is to define a set of convex subsets of the search space. Then, for each subsets the PQP algorithm can efficiently generate an efficient frontier, a so-called envelope. The combination of all generated envelopes then forms the overall solution to the problem.

To our knowledge, our approach is the first metaheuristic approach which is not point-based, but which is capable of generating a continuous front of alternatives for portfolio selection problems with nonconvex constraints. Compared with a state-of-the-art point-based MOEA, E-MOEA was shown to find significantly better frontiers in a shorter time. Both approaches presented in this chapter do not depend on the availability of a highly efficient QP solver, as did the ϵ -Constraint heuristics presented in the previous chapter.

The results presented above showed clearly that the E-MOEA is superior to the P-MOEA with respect to solution quality and calculation time. The only arguments for P-MOEA are flexibility and ease of implementation: (i) When working with the P-MOEA, it is not too problematic to exchange the variance as a measure of dispersion with e.g. Value at Risk. (ii) It is by far easier to implement an algorithm similar to the P-MOEA we have used, but it definitely is not trivial to produce an working version of the E-MOEA, since this additionally requires a fast PQP algorithm and an efficient procedure to calculate the aggregated front.

5 *An Envelope-Based MOEA*

As future work, we are planning to integrate a more intelligent mutation operator that uses shadow prices to influence mutation probabilities. Also, the ideas of envelope-based MOEAs should be transferred to other applications. In particular, the idea seems very promising when applied to portfolio re-balancing problems that include fixed and variable transaction costs.