Chapter V Constrained Ordinal Optimization

We discussed single-objective optimization in Chapter II and III, and dealt with multiple-objective optimization in Chapter IV. All these belong to unconstrained optimizations. Since we usually meet constraints in practice, a natural question is how we could apply ordinal optimization in constrained optimization problems. Traditionally, optimization problems involving constraints are treated via the use of LaGrange multipliers (Bryson and Ho 1969). See also Eq. (4.1) in the introduction of Chapter IV. The duality between constrained optimization with vector optimization is best illustrated via the following diagram (Fig. 5.1).

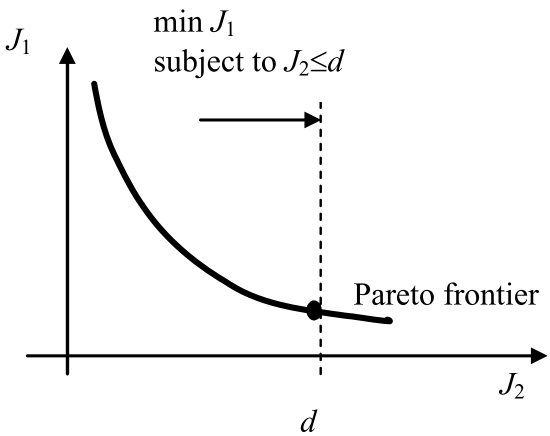


Fig. 5.1. The duality between constrained optimization and vector optimization

Setting different values of the parameter "d", we can determine various points on the Pareto frontier by solving a series of constrained optimization problem. Conversely, every point on the Pareto frontier solves a constrained optimization problem for some constraint value "d". Thus in principle, VOO and COO are also duals of each other. Chapter IV can be considered as a dual of this chapter.

More practically, in some cases, the constraints can be easily checked, e.g., simple linear inequality equations involving one or two variables. By modifying the design space, Θ, to one including only the feasible designs, Θ_f, we convert the original problem to an unconstrained one, which can then be dealt with by the methods in Chapter II-IV. However, in some other cases, it is time-consuming to check the constraints. In this chapter, we focus on the optimization problem of form

$$\begin{aligned} &\min_{\theta\in\Theta} J(\theta) \equiv E\left[L\left(x,\theta,\xi\right)\right] \\ &\text{s.t.} \quad h_i(\theta) \equiv E\left[L_i\left(x,\theta,\xi\right)\right] \le 0, i=1,\dots,m. \end{aligned} \tag{5.1}$$

Considering the problems handled by OO in previous chapters, where the evaluation of $J(\bullet)$ is time consuming, we observe an additional difficulty: there are simulation-based constraints in Eq. (5.1), which makes the precise determination of the feasibility beforehand extremely difficult. In fact, incorporating constraints efficiently is one of the major challenges in developing any simulation-based optimization methods. One naive and impractical approach is to accurately determine the feasibility of a design (this will be referred to as a perfect feasibility model), then apply OO directly within Θ_f, the subset of all the feasible designs. The other extreme is to apply OO directly regardless of the constraints. This does not work in general since many designs in the selected set may be infeasible. The selected set, the size of which is determined without any consideration of the constraints, can no longer ensure to cover some feasible designs with good enough performance with high probability.

The key idea in this chapter is to note, in practice, although we do not have perfect feasibility model, we usually have some rules, experiences, heuristics, and analytical methods (these will be referred as the feasibility models) to help us find feasible designs with a reasonably high probability (certainly no less than 0.5). These feasibility models usually are not perfect, and some times make mistakes, e.g., some designs may be predicted as feasible by the feasibility model, but are actually infeasible. If we incorporate this fact of imperfect (but with some reasonable chance, say 70% or 80%) feasibility prediction into the determination of the size of the selected set, we can ensure to find some feasible and good enough designs with high probability. Before we discuss how to do this incorporation in details, which will be introduced in Section 1, we would like to make some comments.

Recall that the spirit of OO is to ask for good enough with high probability instead of best for sure. The spirit of the above COO is similar: To accommodate the constraints, we ask for feasibility with high probability instead of feasible for sure. It is interesting to note that the classification of "*feasible* vs. *infeasible*" is ***ordinal***. All the advantages of OO apply here, i.e., it can be reasonably easy to obtain a group of truly feasible designs with high probability instead of one for sure. In addition, "*imperfectness*" of the feasibility model is also in tune with the "goal softening" tenet. Although individual determination of feasibility using a crude model may give erroneous results, the model could be very robust with respect to a group of candidates overall. As in the case of the regular OO, $|G \cap S|$ can be good even in the presence of large "noise." The above approach is an evolution of the OO methodology amenable to constrained optimization problems with a "*complete ordinal*" concept. The tenets of "*goal softening*" and "*ordinal comparison*" are reflected by the integration of "imperfect feasibility model" and "feasibility determination."

1 Determination of selected set in COO

As discussed in Chapter III, the effectiveness of the OO technique depends, in part, on the selection rule that we use to select the subset *S*. The simplest selection rule that requires no performance evaluation or estimate is Blind Pick. Analytical results as shown in Chapter II are available for Blind Pick without constraints as in Eq. (5.2),

$$\mathrm{Prob}\left[\left|G \cap S\right| \geq k\right] = \sum_{i=k}^{\min(g,s)} \frac{\binom{g}{i}\binom{N-g}{s-i}}{\binom{N}{s}}. \tag{5.2}$$

Here we will derive analytical results for the constrained cases. When we have constraints, since there are infeasible designs in Θ, if we still use Blind Pick, we will have to select more designs to guarantee the same level of alignment. It is also reasonable to see that the required size of the selected set decreases as the predication accuracy increases.

1.1 Blind pick with an imperfect feasibility model

In engineering practice, we usually have an imperfect feasibility model, which is based on rules, experiences, heuristics, and some analytical methods that can be easily checked. Such a model can make prediction about the feasibility of a design choice θ with little or no computation. However, its prediction will sometimes be faulty. Suppose we first use this feasibility model to obtain N (say 1000) designs from the entire design space Θ as follows. We uniformly sample Θ and test the feasibility with our feasibility model, then accept designs predicted as feasible and reject designs predicted as infeasible. We denote such set of designs as $\hat{\Theta}_f$. Then we apply BP within $\hat{\Theta}_f$ to select a subset S_f. We want to find some truly good enough and feasible deigns of Θ. The rationale here is as follows: when the set of predicted feasible designs $\hat{\Theta}_f$ is large, the density ρ_f of feasible designs in the design space is reasonably high (say no less than 10% of the entire design space)[1], and the feasibility model has a reasonable accuracy (with probability no less than 0.5 to give correct prediction), there should be some truly good enough and feasible designs of Θ contained in $\hat{\Theta}_f$. We call this method Blind Pick with a (imperfect) Feasibility Model (BPFM).

In order to quantify the alignment probability, we denote the set of top-$100 \times \alpha_g$% truly feasible designs in $\hat{\Theta}_f$ as G, the good enough set. Suppose there are N_f truly feasible designs in the N predicted feasible designs[2]. Then the size of the good enough set $g = N_f \alpha_g$. We use P_{e1} and P_{e2} to measure the accuracy of a feasibility model, i.e., P_{e1} denotes the probability that a truly feasible design is predicted as infeasible, also known as the type-I error; and P_{e2} denotes the probability that a truly infeasible design is predicted as feasible, also known as the type-II error. To simplify the discussion, let us assume $P_{e1} = P_{e2}$ first, remove this constraint, and discuss the more general case later. Let $P_f = 1\text{-}P_{e1}$, then P_f is the prediction accuracy of the feasibility model, that is, the probability that a design is predicted as feasible if it is truly feasible and also the probability that a design is predicted as infeasible if it is truly infeasible. So, for each θ design in $\hat{\Theta}_f$, which is predicted as feasible, the probability that it is truly feasible can be obtained via Bayesian formula

[1] When the density of feasible designs is much less than 10%, we need to improve the value of N or use a good feasibility model so that there are some truly good enough and feasible designs of Θ contained in $\hat{\Theta}_f$.

[2] The selection of N should guarantee N_f is large enough.

$$\begin{aligned} r &= \text{Prob}\left[\theta \text{ is feasible}/\theta \text{ is predicted as feasible}\right] \\ &= \frac{\text{Prob}\left[\theta \text{ is feasible and } \theta \text{ is predicted as feasible}\right]}{\text{Prob}\left[\theta \text{ is predicted as feasible}\right]} \\ &= \frac{\text{Prob}\left[\theta \text{ is feasible}\right]\text{Prob}\left[\theta \text{ is predicted as feasible}/\theta \text{ is feasible}\right]}{\text{Prob}\left[\theta \text{ is predicted as feasible}\right]} \\ &= \frac{\rho_f P_f}{\text{Prob}\left[\theta \text{ is predicted as feasible}\right]} \\ &= \frac{\rho_f P_f}{\rho_f P_f + \left(1-\rho_f\right)\left(1-P_f\right)}. \end{aligned} \tag{5.3}$$

Thus, because the feasibility model is usually not perfect (that is P_f<1), infeasible designs cannot be completely excluded in S_f.

It should be pointed out that although the expected number of feasible designs in $\hat{\Theta}_f$ is $N_f = N \times r$ and the expected number of feasible designs in S_f is $\left|S_f\right| \times r$, the results of the regular unconstrained OO method cannot be directly applied in this case with $s = \left|S_f\right| \times r$.

Exercise 5.1: Explain intuitively why?

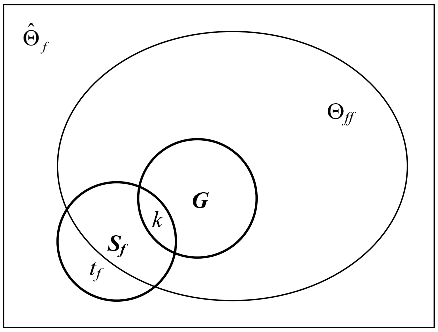
We shall now derive the AP of the selected subset S_f by averaging all possible numbers of feasible designs in S_f.

Suppose the size of selected subset S_f is s_f. The number of infeasible designs in the selected subset S_f, denoted as t_f, follows approximately a *Binomial distribution*, i.e., $t_f \sim b(s_f, r)$, where the size of selected subset s_f is the size of the *Bernoulli trials* and r is the probability that a selected design in S_f is feasible. The probability that there are $t_f = j$ infeasible designs in selected subset S_f, is:

$$\text{Prob}\left[t_f = j\right] = \binom{s_f}{j} (r)^{s_f - j} (1-r)^j. \tag{5.4}$$

Given that there are t_f *infeasible designs in the selected subset* S_f, the conditional AP that there are exact k good enough designs in S_f is given by:

$$\text{Prob}\left[\left|G \cap S_f\right| = k / t_f\right] = \frac{\binom{g}{k}\binom{N_f - g}{s_f - t_f - k}}{\binom{N_f}{s_f - t_f}}. \tag{5.5}$$



$\hat{\Theta}_f$: the set of predicted feasible designs
Θ_{ff} : the set of truly feasible designs in $\hat{\Theta}_f$
$\boldsymbol{G}$: the set of good enough designs in $\hat{\Theta}_f$
$\boldsymbol{S_f}$: the selected set
k : the alignment level
t_f : the number of infeasible designs in S_f

Fig. 5.2. Illustration of Eq. (5.5)

Eq. (5.5) is a direct analog of Eq. (5.2), which was first derived in Eq. (2.37). Please see also Fig. 5.2 for illustration.

Since if there are k feasible and good enough designs in S_f, the number of infeasible designs in S_f could be any number from 0 to $\min(s_f - k, N\text{-}N_f)$, based on the Total-Probability Theorem, we have the formula for the AP that there are at least k good enough designs in the selected set S_f as

$$\begin{aligned} \mathrm{AP}_{\mathrm{COO}} &= \mathrm{Prob}\left[\left|G \cap S_f\right| \geq k\right] \\ &= \sum_{i=k}^{\min(g,s_f)} \sum_{j=0}^{\min(s_f-i,N-N_f)} \frac{\binom{g}{i}\binom{N_f-g}{s_f-j-i}}{\binom{N_f}{s_f-j}} \binom{s_f}{j} (r)^{s_f-j} (1-r)^{j}. \end{aligned} \tag{5.6}$$

If we do not have any knowledge about the feasibility of the designs, each of them is equally likely to be feasible or infeasible. This corresponds to the special case where $P_f = 0.5$ and thus $r = \rho_f$ in Eq. (5.6). It is also interesting to note that, if we have perfect knowledge about the feasibility of each design, by sampling only truly feasible designs, we can obtain an $\hat{\Theta}_f$ containing N feasible designs. This corresponds to the special case $P_f = 1$ and thus $r = 1$ in Eq. (5.6). Direct calculation shows that Eq. (5.6) reduces to Eq. (5.2) if $P_f = 1$ and thus $r = 1$.

1.2 Impact of the quality of the feasibility model on BPFM

The value of BPFM lies in the fact that, by only very crude feasibility model, we can bring an impressive improvement to the efficiency of COO. First, we will show that AP_{COO} is an increasing function of P_f, the accuracy of feasibility model and also an increasing function of ρ, the density of feasible designs in the entire design space. We show this result through two steps. In the first step, we show that AP_{COO} is an increasing function of r, and in the second step, we show that r is an increasing function of P_f and ρ_f. Since r represents the probability that an observed feasible design is truly feasible, the number of truly feasible designs on average in $\hat{\Theta}_f$ is Nr, i.e., $N_f = Nr$. Since the good enough set G is defined as the top-$100 \times \alpha_g$% of these N_f truly feasible designs, $g = N_f \alpha_g = N\alpha_g r$. Thus, when r increases, g increases. Since we are doing blind picking in $\hat{\Theta}_f$, all other parameters remaining the same (i.e., fixing N and s_f), the AP Prob[$|G \cap S_f| \geq k$] increases. Now, we show that r is an increasing function of P_f and ρ_f. Fix ρ_f, following Eq. (5.3), we have that

$$\frac{dr}{dP_f} = \frac{\rho_f\left(1-\rho_f\right)}{\left(\rho_f P_f + \left(1-\rho_f\right)\left(1-P_f\right)\right)^2} > 0 \text{, for all } 0 < \rho_f < 1.$$

This shows that r is an increasing function of P_f. Similarly we can show r is an increasing function of ρ_f. In total, we show that the AP Prob[$|G \cap S_f| \geq k$] is an increasing function of P_f and ρ_f, which is also intuitively reasonable.

Then, we will show some numerical results. Suppose the size of design space $\hat{\Theta}_f$ is 1000. The number of feasible designs is $N_f = 500$. The good enough set G is the top 50 feasible designs (i.e., the top-10% feasible designs in $\hat{\Theta}_f$). The AP versus the size of selected subset, s_f is plotted in Fig. 5.3. As expected, for the constrained problem, the BPFM method with a feasibility model P_f>0.5 is better than that without a feasibility model (i.e., $P_f = 0.50$, which is identical to directly using BP), because, for the same size of the selected subset, the AP obtained for P_f>0.50 is larger than that obtained for $P_f = 0.50$. It is also observed that the more accurate the feasibility model (larger P_f) is, the higher AP we can achieve.

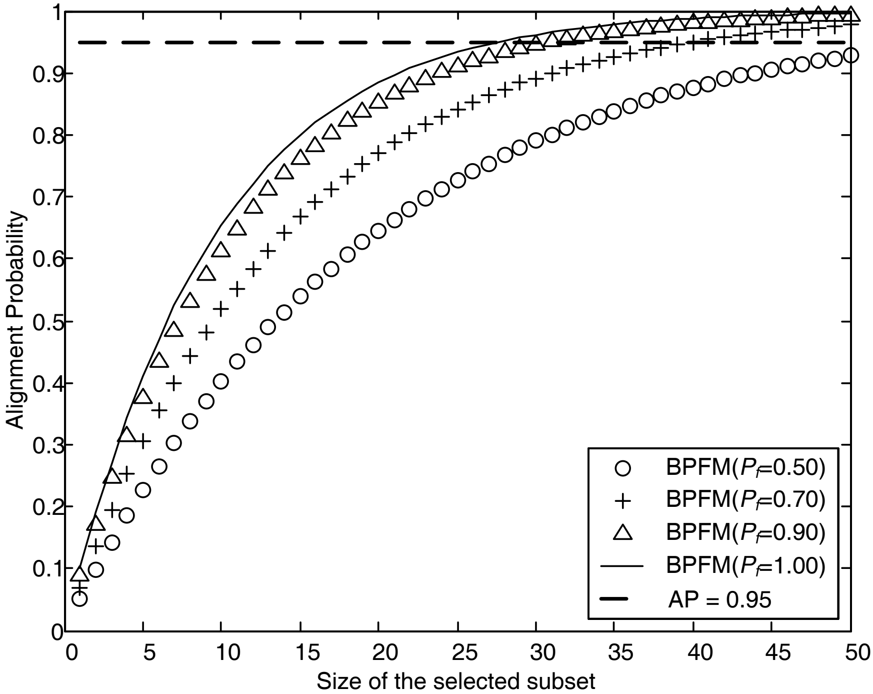


Fig. 5.3. AP versus the subset selection size of BP and BPFM

The sizes of selected subsets obtained by different P_f values are illustrated in Table 5.1. It is shown that, for the same required AP, a larger P_f requires a smaller selected subset, and thus is more efficient. In other words, a smaller selected subset is required for a more accurate feasibility model, for a given level of alignment probability.

Table 5.1. Sizes of the selected subsets

Required AP	$P_f = 0.50$	$P_f = 0.70$	$P_f = 0.90$	$P_f = 1.00$
≥0.50	14	10	8	7
≥0.60	18	13	10	9
≥0.70	24	17	13	12
≥0.80	31	22	17	16
≥0.90	44	32	24	22
≥0.95	57	41	31	28
≥0.99	87	61	47	42

So far we assume that $P_{e1} = P_{e2}$ to simplify the notation. Now, we show how to remove this constraint and consider the more general case where r is a function of P_{e1} and P_{e2}. Following a similar analysis to Eq. (5.3), we have

$$r = \frac{\rho_f (1-P_{e1})}{\rho_f (1-P_{e1}) + (1-\rho_f) P_{e2}}. \tag{5.7}$$

Giving a feasibility model, once we estimate the accuracy of the feasibility model, i.e., P_{e1} and P_{e2}, we can use Eq. (5.7) to calculate r and then use Eq. (5.6) to quantify the AP_{COO}. We now show when the accuracy of the feasibility model increases, i.e., P_{e1} and P_{e2} decreases, r increases, and then following the analysis similar to the beginning of this subsection, we can see that AP also increases. Fix P_{e2} and ρ_f, we have

$$\frac{dr}{dP_{e1}} = -\frac{\rho_f (1-\rho_f) P_{e2}}{\left(\rho_f (1-P_{e1}) + (1-\rho_f) P_{e2}\right)^2} < 0.$$

Similarly, fix P_{e1} and ρ_f, we have

$$\frac{dr}{dP_{e2}} = -\frac{\rho_f (1-\rho_f)(1-P_{e1})}{\left(\rho_f (1-P_{e1}) + (1-\rho_f) P_{e2}\right)^2} < 0.$$

And fix P_{e1} and P_{e2}, we have

$$\frac{dr}{d\rho_f} = \frac{(1-P_{e1}) P_{e2}}{\left(\rho_f (1-P_{e1}) + (1-\rho_f) P_{e2}\right)^2} > 0.$$

This means r is a decreasing function of P_{e1} and P_{e2}, and an increasing function of ρ_f. The previous discussion on $P_{e1} = P_{e2} = 1\text{-}P_f$ is a special case.

Suppose we are given a feasibility model which predicts the feasibility of a design accurately with probability P_f. We summarize the application procedure of COO using this feasibility model as follows.

Box 5.1. COO approach

Step 1. Find a feasibility model and randomly sample N predicted feasible designs.
Step 2. Specify g and k.
Step 3. Estimate ρ_f, the density of feasible designs in the entire design space and estimate the accuracy of the feasibility model, i.e., the P_{e1} and P_{e2}, and calculate r through Eq. (5.7).
Step 4. Apply the BPFM in Eq. (5.6) to determine the size of the selected set.
Step 5. Randomly select S_f designs from the N designs.
Step 6. The COO theory ensures that there are no less than k good enough feasible designs in the selected subset with high probability.

Exercise 5.2: How can we determine the size of the selected set if we use Horse Race instead of Blind Pick in Step 4 above within the set of predicted feasible designs?

2 Example: Optimization with an imperfect feasibility model

In this section, we use a simple example to evaluate the effects of COO under different observation noise. As expected, since we are developing blind pick based COO, the alignment level of selected set should be insensitive to the level of noise in observation. Let us consider the following constrained optimization problem. Each design θ is an integer between 1 and 1000, i.e., $\Theta=\{1,2,\ldots 1000\}$. The objective function $J(\theta) = \theta$. The constraint is that θ must be even numbers. This problem can then be mathematically formulated as

$$\begin{aligned} &\min_{\theta\in\{1,2,\ldots 1000\}} J(\theta)=\theta \\ &\text{s.t. } \operatorname{mod}(\theta,2)=0 \end{aligned} \tag{5.8}$$

where mod(•,•) is the modulo operator. Suppose also observation noise ξ contains i.i.d. uniform distribution $U(0,a)$ such that our observation is

$$\hat{J}(\theta) \equiv J(\theta) + \xi \, .$$

The presence of noise makes the optimization problem non-trivial to solve even with perfect knowledge about the feasibility of each design.

Suppose we also have an imperfect feasibility model, which gives the correct feasibility prediction with probability P_f. In other words, when the design θ is an even number (i.e., a truly feasible design), the feasibility model predicts the design as feasible with probability P_f (thus predicts the design as infeasible with probability $P_{e1} = 1\text{-}P_f$); when the design θ is an odd number (i.e., a truly infeasible design), the feasibility model predicts the design as infeasible with probability P_f (thus predicts the design as feasible with probability $P_{e2} = 1\text{-}P_f$). We reasonably assume $P_f \geq 0.5$ (Otherwise we simply reverse the prediction given by this feasibility model, and can then obtain a "reasonable" feasibility model).

Suppose we want to find at least one of the truly top 50 feasible designs with high probability, i.e., $g = 50$, $k = 1$ with $G = \{\boldsymbol{2},\boldsymbol{4},\boldsymbol{6},\ldots\boldsymbol{100}\}$. We simulate the BPFM method with $P_f = 0.95$. Then $P_{e1} = P_{e2} = 0.05$. Notice for this example, half of the designs are feasible, so we have $\rho_f = 0.5$. By Eq. (5.7), we calculate that $r = 0.95$. The size of the selected subset, s_f, then can be calculated based on Eq. (5.6) with the different required AP (0.50, 0.70, and 0.95). The selected subsets S_f for different AP are shown in Table 5.2. It turns out that the BPFM method finds at least one of the good enough feasible designs in all the instances as shown.

Table 5.2. A random examination of BPFM ($P_f = 0.95$)

Required AP	s_f	Selected subset S_f	Alignment level
≥0.50	7	{808, 524, ***32***, 850, 240, 498, 878}	1
≥0.70	12	{714, 284, 982, 614, 644, 972, 238, 820, 986, 176, 272, ***30***}	1
≥0.90	23	{350, 490, 760, 147, 236, 483, ***88***, 130, 260, 456, ***24***, 508, 997, 178, 228, 564, 842, 976, 446, 660, 330, 952, 87}	2
≥0.95	30	{842, 812, 716, 682, 980, ***8***, 510, 272, 996, 588, 410, 718, 154, 427, 964, 806, 558, 502, 414, 724, 998, 265, 384, 772, 262, 682, 572, 990, 564, 626}	1

Note that the good results in Table 5.2 are not coincidences since the BPFM method blindly picks designs from the space that are predicted as feasible by the feasibility model without relying on accurate estimation on $J(\theta)$. So, we can expect that the guarantee provided by the BPFM method holds no matter how large the noise is.

3 Conclusion

Optimization of DEDS with complicated stochastic constraints is generally very difficult and simulation is usually the only way available. The results on unconstrained OO in Chapter II cannot be applied directly since many infeasible designs cannot be excluded without costly simulation. The COO approach with feasibility model presented in this chapter is effective to solve this long-standing problem. According to No-Free-lunch Theorem (Ho et al. 2003), *no algorithm can do better on the average than blind search without structural information.* The feasibility model in this case can be regarded as the "structural information." As a result, COO provides a more efficient approach for solving constrained optimization problems, since the size of the selected subset is smaller than that when directly applying the unconstrained OO approach.

The algorithm for subset selection and the procedure of *Blind Pick with Feasibility Model* (BPFM) for COO are derived. Numerical testing shows that, by using COO method, to meet the same required alignment probability, *Blind Pick with Feasibility Model* is more efficient than pure Blind Pick. The testing results also show that the method is very robust, even when the feasibility model is not very accurate. Furthermore, the COO method presented in this chapter is a general approach. Any crude feasibility model even with large noise is compatible and can work well with the approach. In Chapter VIII Section 3, we apply COO with a feasibility model based on the rough set theory to a real world remanufacturing system, and yields promising results. ***Similarly, the application of this approach of COO is not restricted to the BP selection method. Other selection methods such as the Horse Race method can also be used in connection with the crude feasibility model. The modifications to the AP of course must be carried out similar to that of Eq. (5.6) except via simulation.*** A quick-and-dirty first approximation is to simply modify the unconstrained UAP by r.