

Trabalho Computacional 3 - Teoria da Decisão (ELE088)

Daniel Felipe de Almeida Araújo
Universidade Federal de Minas Gerais
Matrícula: 2023422617

Milton Pereira Bravo Neto
Universidade Federal de Minas Gerais
Matrícula: 2018072549

Raphael Henrique Braga Leivas
Universidade Federal de Minas Gerais
Matrícula: 2020028101

Resumo—Este trabalho busca resolver um problema multiobjetivo de uma empresa explorando-o por partes, com a implementação do algoritmo BVNS (*Basic Variable Neighborhood Search*) para otimização mono-objetivo, comparação das formulações ϵ -Restrito e soma ponderada para o problema multiobjetivo na formação da fronteira Pareto. Foi utilizado ainda o indicador de qualidade HVI para uma análise do conjunto de soluções pareto-ótimas, e comparados os métodos AHP e PROMETHEE para a realização da tomada de decisão.

I. INTRODUÇÃO

É apresentado um problema multiobjetivo que busca gerenciar as manutenções dos equipamentos de uma empresa, separados em 4 *clusters* ou grupos, de forma a otimizar simultaneamente o custo total envolvido na realização das manutenções e o custo total em caso de falha dos equipamentos.

Existem três tipos de manutenção que podem ser aplicadas em cada equipamento, e cada grupo possui uma função que descreve o custo de falha de seus equipamentos. São fornecidas todas as informações relativas à modelagem matemática dos custos de manutenção e falha, porém as características e funcionalidades reais dos equipamentos e qual o critério de agrupamento permanecem desconhecidos.

No roteiro apresentado para a exploração do problema são passadas orientações e critérios que idealmente devem ser seguidos, para uma melhor abordagem na resolução que levará à tomada de decisão. É indicada uma primeira análise dos problemas de otimização numa perspectiva mono-objetiva isolando os problemas, para assim ser desenvolvido um algoritmo que os resolva. Depois, explora-se as opções de formulações para a resolução do problema multiobjetivo definindo-se uma fronteira Pareto-ótima com as possíveis soluções candidatas, e por fim, são aplicadas duas estratégias para a tomada de decisão.

II. METODOLOGIA

A. Modelagem do Problema

Inicialmente podemos ver o trabalho como sendo dois problemas mono-objetivo distintos:

- Problema 1: minimização do custo de manutenção total $f_1(\cdot)$
- Problema 2: minimização do custo esperado de falha total $f_2(\cdot)$

1) *Problema 1*: Temos essencialmente um problema de designação simples. Seja N o número de equipamentos e J o número de políticas de manutenção, definimos a variável de decisão x_{ij} por

$$x_{ij} : \text{se o equipamento } i \text{ executa a manutenção } j \quad (1)$$

onde

$$x_{ij} \in \{0, 1\} \quad , \quad i = \{1, 2, \dots, N\} \quad , \quad j = \{1, 2, \dots, J\}$$

Para a função objetivo, seja c_j o custo de executar a manutenção j . Note que esse custo independe do equipamento i que estamos executando a manutenção. Temos a função objetivo

$$\min f_1 = \sum_{i=1}^N \sum_{j=1}^J c_j x_{ij} \quad (2)$$

sujeito a

$$\sum_{j=1}^J x_{ij} = 1 \quad , \quad \forall i = 1, 2, \dots, N \quad (3)$$

A 3 indica que todo equipamento executa exatamente uma política de manutenção. Além disso, note que solução da 2 é trivial: basta escolher o plano de manutenção com o menor custo para todos os equipamentos.

2) *Problema 2*: O custo da falha de cada equipamento é dado pelo produto da probabilidade de falha p_{ij} pelo custo da falha do equipamento, dada por d_i . Assim, temos

$$\min f_2 = \sum_{i=1}^N \sum_{j=1}^J p_{ij} d_i x_{ij} \quad (4)$$

onde

$$x_{ij} \in \{0, 1\} \quad , \quad i = \{1, 2, \dots, N\} \quad , \quad j = \{1, 2, \dots, J\}$$

$$p_{ij} = \frac{F_i(t_0 + k_j \Delta t) - F_i(t_0)}{1 - F_i(t_0)} \quad (5)$$

$$F_i(t) = 1 - \exp \left[- \left(\frac{t}{\eta_i} \right)^{\beta_i} \right] \quad (6)$$

sujeito a

$$\sum_{j=1}^J x_{ij} = 1 \quad , \quad \forall i = 1, 2, \dots, N \quad (7)$$

Note que na Equação 4 temos essencialmente um problema de programação linear inteira. Assim, é possível usar o método Simplex visto em Pesquisa Operacional para resolver esse problema com garantia de otimalidade. Usando o Simplex, a solução encontrada foi

$$\mathbf{x}^* = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 1 \end{bmatrix} \quad , \quad f_2(\mathbf{x}^*) = 1048.17$$

Assim, antes mesmo de começar a implementar o BVNS para resolver os problemas isoladamente, já sabemos as soluções ótimas para eles.

3) *Modelagem Multiobjetivo*: Juntando as modelagens dos problemas mono-objetivos acima, temos a modelagem multi-objetivo do problema.

$$\begin{aligned} \min f_1 &= \sum_{i=1}^N \sum_{j=1}^J c_j x_{ij} \\ \min f_2 &= \sum_{i=1}^N \sum_{j=1}^J p_{ij} d_i x_{ij} \end{aligned}$$

x_{ij} : se o equipamento i executa a manutenção j

sujeito a

$$\sum_{j=1}^J x_{ij} = 1 \quad , \quad \forall i = 1, 2, \dots, N$$

$$x_{ij} \in \{0, 1\} \quad , \quad i = \{1, 2, \dots, N\} \quad , \quad j = \{1, 2, \dots, J\}$$

onde

- $N = 500$: número de equipamentos
- $J = 3$: número de planos de manutenção
- c_j : custo de executar a manutenção j
- p_{ij} : probabilidade de falha do equipamento i executando a manutenção j , definido em 5 e 6
- d_i : custo de falha do equipamento i

A partir dessa modelagem, temos o nosso problema multi-objetivo

$$\min \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x})] \quad (8)$$

Considerando o problema de (8), podemos aplicar duas abordagens escalares para obter a fronteira Pareto-ótima no espaço de objetivos, descritas a seguir.

B. Formulações para resolução do problema multiobjetivo

1) *Formulação Soma Ponderada* P_w : Seja $0 \leq w \leq 1$ um peso qualquer gerado aleatoriamente de uma distribuição uniforme no intervalo $[0, 1]$. Usando a abordagem da soma ponderada, podemos reescrever (8) na forma de mono-objetivo de

$$\min f_w = \min w f_1 + (1 - w) f_2 \quad (9)$$

onde (9) está sujeito às mesmas restrições do problema original. Como (9) é escalar, podemos minimizar f_w através de métodos já conhecidos como o Simplex e o BVNS.

2) *Formulação ϵ -Restrito* P_ϵ : Com a abordagem do ϵ -Restrito, vamos minimizar apenas f_1 usando f_2 como restrição. Seja ϵ_2 um real qualquer tal que $\min f_2 \leq \epsilon_2 \leq \max f_2$. Temos

$$\min f_1 \quad (10)$$

sujeito a

$$\begin{cases} f_2 \leq \epsilon_2 \\ \sum_{j=1}^J x_{ij} = 1 \quad , \quad \forall i = 1, 2, \dots, N \end{cases} \quad (11)$$

em que (10) possui as mesmas restrições do problema original mais a restrição de $f_2 \leq \epsilon_2$.

Contudo, como o BVNS é usado para resolver problemas de otimização irrestritos, precisamos converter (10) em um problema irrestrito. Para isso, adicionamos o termo um termo de penalidade $p(x, u)$ da seguinte forma:

$$p(x, u) = u \max[0, g(x)]^2$$

onde $g(x)$ é a nossa restrição de desigualdade, dada por

$$g(x) \leq 0 \implies f_2 - \epsilon_2 \leq 0 \implies g(x) = f_2 - \epsilon_2$$

de modo que o nosso problema irrestrito se torna:

$$\min f_1 + u \max[0, f_2 - \epsilon_2]^2 \quad (12)$$

Note que as demais restrições já estão naturalmente incluídas no BVNS devido à maneira como nós fizemos a representação computacional das variáveis de decisão, de modo que só precisamos fazer a correção para a restrição do ϵ em (12).

3) *Normalização*: Para garantir que as abordagens escalares sejam condizentes, precisamos normalizar f_1 e f_2 através de

$$f_1(\mathbf{x}) = \frac{f_1(\mathbf{x}) - \min f_1}{\max f_1 - \min f_1} \quad , \quad f_2(\mathbf{x}) = \frac{f_2(\mathbf{x}) - \min f_2}{\max f_2 - \min f_2} \quad (13)$$

de modo que a formulação da soma ponderada de (9) pode ser reescrita como

$$\min \left(w \frac{f_1(\mathbf{x}) - \min f_1}{\max f_1 - \min f_1} + (1 - w) \frac{f_2(\mathbf{x}) - \min f_2}{\max f_2 - \min f_2} \right)$$

que será usado como função objetivo no código do BVNS.

4) *BVNS - Basic Variable Neighborhood Search*: O Algoritmo 1 mostra a versão do BVNS implementada no trabalho.

Algoritmo 1 BVNS implementado no trabalho.

```

1: procedure BVNS( $\mathbf{x}$ ,  $k_{\max}$ )
2:   while num_sol_avaliables < max_sol_avaliables do
3:      $k \leftarrow 1$ 
4:     while  $k < k_{\max}$  do
5:        $\mathbf{x}' \leftarrow \text{SHAKE}(\mathbf{x}, k)$ 
6:        $\mathbf{x}'' \leftarrow \text{FIRSTIMPROVEMENT}(\mathbf{x}, \mathbf{x}', k)$ 
7:        $\mathbf{x}, k \leftarrow \text{NEIGHBORHOODCHANGE}(\mathbf{x}, \mathbf{x}'', k)$ 
8:     end while
9:   end while
10: end procedure

```

O Algoritmo 2 mostra a função Shake. Nela estão definidas as três estruturas de vizinhança escolhidas para implementação. As duas primeiras são vizinhanças de refinamento, mas com abordagens diferentes. E a terceira é uma vizinhança de perturbação para buscar sair de mínimos locais:

- A primeira estrutura é o que chamamos de um movimento de *1-swap*, onde é escolhido aleatoriamente um equipamento e trocado seu plano para um dos outros dois restantes, a escolha do plano também é aleatória.
- A segunda estrutura é a troca ou permutação dos planos de dois equipamentos diferentes escolhidos também aleatoriamente.
- A terceira estrutura por sua vez, altera um bloco de 50 equipamentos em sequência, onde o início do bloco é aleatório. Nesse bloco é avaliado qual o plano mais comum e troca-se o plano de manutenção de todos os integrantes do bloco para um mesmo plano, diferente do mais comum encontrado anteriormente.

Algoritmo 2 Função Shake.

▷ Gera uma solução aleatória na k -ésima estrutura de vizinhança.

```

1: procedure SHAKE( $\mathbf{x}$ ,  $k$ )
2:   if  $k = 1$  then
3:      $\mathbf{y} \leftarrow 1\text{-swap}$ 
4:   end if
5:   if  $k = 2$  then
6:      $\mathbf{y} \leftarrow \text{Permutação de dois planos de manutenção}$ 
7:   end if
8:   if  $k = 3$  then
9:      $\mathbf{y} \leftarrow \text{Mudança de um bloco de equipamentos para outro plano}$ 
10:  end if
11:  return  $\mathbf{y}$ 
12: end procedure

```

5) *Estratégias de Refinamento*: O Algoritmo 3 mostra a função de busca local implementada após gerar uma solução

aleatória com o Shake. Ela basicamente realiza uma busca em até $N = 100$ vizinhos à solução inicial \mathbf{x}' do Shake, e retorna a primeira solução \mathbf{x}'' cujo valor da solução objetivo é menor do que o valor do objetivo na solução inicial \mathbf{x}' do Shake. Caso nenhuma solução melhor é encontrada, retorna a solução inicial \mathbf{x}' .

Algoritmo 3 Função FirstImprovement.

▷ Busca uma primeira solução na vizinhança de \mathbf{x}' melhor que \mathbf{x}' .

```

1: procedure FIRSTIMPROVEMENT( $\mathbf{x}', k$ )
2:    $N \leftarrow 100$ 
3:   for all  $i$  in range( $N$ ) do
4:      $\mathbf{x}'' \leftarrow \text{SHAKE}(\mathbf{x}', k)$ 
5:     if  $f(\mathbf{x}'') < f(\mathbf{x}')$  then
6:       return  $\mathbf{x}''$ 
7:     end if
8:   end for
9:   return  $\mathbf{x}'$ 
10: end procedure

```

É possível fazer uma pequena modificação no Algoritmo 3 para obter o Best Improvement, exibido no Algoritmo 4. Note que essa função sempre executa as N buscas por uma melhor solução, e portanto o código é mais caro computacionalmente que no Algoritmo 3. No entanto, em geral, a solução encontrada pelo BestImprovement será melhor que a do FirstImprovement.

Algoritmo 4 Função BestImprovement.

▷ Busca a melhor solução na vizinhança de \mathbf{x}' melhor que \mathbf{x}' .

```

1: procedure BESTIMPROVEMENT( $\mathbf{x}', k$ )
2:    $N \leftarrow 100$ 
3:    $\mathbf{x\_melhor} \leftarrow \mathbf{x}'$ 
4:   for all  $i$  in range( $N$ ) do
5:      $\mathbf{x}'' \leftarrow \text{SHAKE}(\mathbf{x}', k)$ 
6:     if  $f(\mathbf{x}'') < f(\mathbf{x\_melhor})$  then
7:        $\mathbf{x\_melhor} \leftarrow \mathbf{x}''$ 
8:     end if
9:   end for
10:  return  $\mathbf{x\_melhor}$ 
11: end procedure

```

6) *Heurística Construtiva*: O Algoritmo 5 mostra a heurística construtiva utilizada para a criação da solução inicial. Basicamente o problema se reduz em escolher um plano de manutenção, dentre os três disponíveis, para cada equipamento minimizando o custo da manutenção e o custo de falha dos equipamentos. A minimização do custo da manutenção se dá escolhendo a manutenção mais barata para todos os equipamentos, e a minimização do custo de falha escolhendo a manutenção mais cara.

Olhando para o segundo problema, temos a matriz de custos de falha $p_{ij}d_i$ onde i é cada equipamento e j os planos de

manutenção. Para cada equipamento i fixo avalia-se a variância de $p_{ij}d_i$ e caso esse valor seja maior que o limiar de 0.5 escolhe a manutenção mais cara para compor a solução inicial daquele equipamento, caso seja menor que o limiar é escolhida a manutenção mais barata.

A lógica envolvida é que se o custo de falha não varia tanto para aquele equipamento, não é necessário a manutenção mais cara.

Algoritmo 5 Heurística construtiva para gerar a solução inicial.

```

1: procedure SOLUCAOINICIAL()
2:    $\mathbf{x} \leftarrow$  Solução aleatória
3:   for all  $i$  in  $\mathbf{x}$  do
4:     if  $\text{variancia}(p_{ij}d_i) \geq \text{limiar}$  then
5:        $\mathbf{x}[i] \leftarrow$  Manutenção mais cara
6:     else
7:        $\mathbf{x}[i] \leftarrow$  Manutenção mais barata
8:     end if
9:   end for
10:  return  $\mathbf{x}$ 
11: end procedure

```

C. Indicador de Hipervolume (HVI)

1) *Formulação Matemática:* Dado um conjunto $S = \{x_1, x_2, \dots, x_n\}$ de soluções não-dominadas em um problema de minimização com m objetivos, e um vetor de referência $z^{\text{ref}} = (z_1^{\text{ref}}, \dots, z_m^{\text{ref}})$, o hipervolume $HV(S)$ é definido como:

$$HV(S) = \text{Vol} \left(\bigcup_{x \in S} [f_1(x), z_1^{\text{ref}}] \times \dots \times [f_m(x), z_m^{\text{ref}}] \right) \quad (14)$$

No caso biobjetivo considerado neste trabalho ($m = 2$), o hipervolume corresponde à área total dominada pelas soluções de S até o ponto de referência z^{ref} .

2) *Interpretação Geométrica:* Geometricamente, o hipervolume representa a união de retângulos formados por cada solução não-dominada e o vetor de referência. Assim, quanto maior o hipervolume, maior a área do espaço dos objetivos que é coberta pelas soluções, indicando tanto boa convergência quanto boa distribuição.

Na Figura 1 temos uma ilustração da métrica de hipervolume para um problema de otimização com duas funções objetivo.

O hipervolume do conjunto de referência (em vermelho, com o ponto de referência $\mathbf{r} = [1, 1]$) é utilizado como fator de normalização.

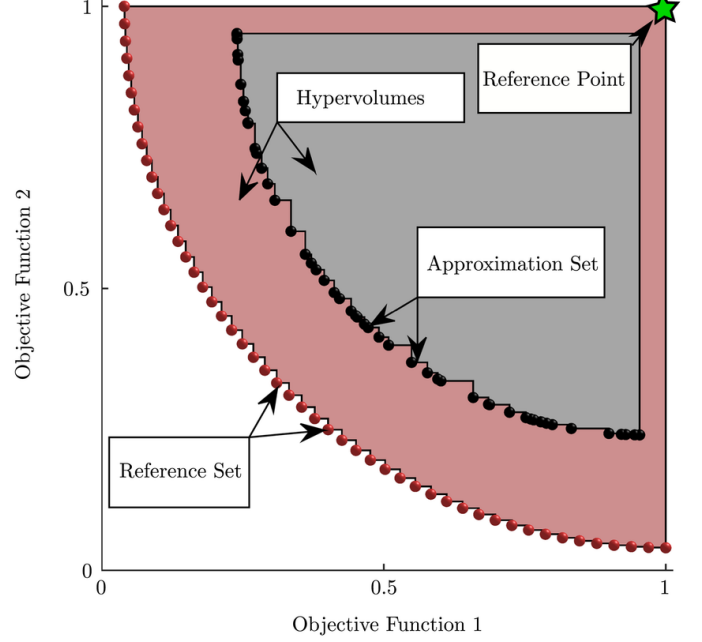
Assim, o hipervolume normalizado do conjunto de aproximação (em cinza) é calculado como:

$$HV_{\text{NAS}} = \frac{HV_{\text{AS}}}{HV_{\text{RS}}}$$

onde:

- HV_{NAS} : hipervolume normalizado do conjunto de aproximação (Normalized Approximation Set)

Fig. 1. Figura ilustrativa da métrica de hipervolume. Fonte: ResearchGate



- HV_{AS} : hipervolume do conjunto de aproximação (Approximation Set)
- HV_{RS} : hipervolume do conjunto de referência (Reference Set)

3) *Implementação:* Para a utilização dessa métrica para avaliar os resultados obtidos pelas fronteiras encontradas, foi fornecido uma função Matlab já ajustada ao problema trabalhado, sendo necessário apenas a entrada de um arquivo “.csv” com 500 colunas e N linhas, onde N são as soluções encontradas. Conforme sugerido nas orientações do trabalho, foi feita a união dos resultados da 5 fronteiras encontradas com 20 pontos cada, e portanto foi passado um arquivo “.csv” com 500 colunas e 100 linhas para cada um dos métodos. O algoritmo fornecido já possui uma filtragem das soluções enviadas, logo não foi necessário uma limpeza de soluções repetidas no arquivo pois isso já era feito automaticamente.

III. RESULTADOS

A. Problemas mono objetivos

A partir da execução do algoritmo BVNS para a resolução dos problemas mono-objetivo foi obtido os seguintes valores:

$$\min f_1 = 0 \quad (15)$$

$$\max f_1 = 1000 \quad (16)$$

$$\min f_2 = 1048.17 \quad (17)$$

$$\max f_2 = 1745.49 \quad (18)$$

Esses resultados são utilizados para a realizar a normalização dos valores ao se utilizar a formulação de Soma Ponderada e os mínimos estão ilustrados nas Figuras 2 e 3.

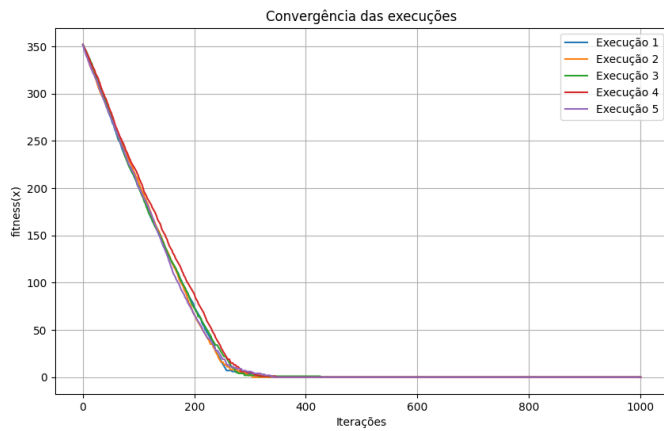


Fig. 2. Convergência do BVNS implementado para o Problema 1 Isolado.

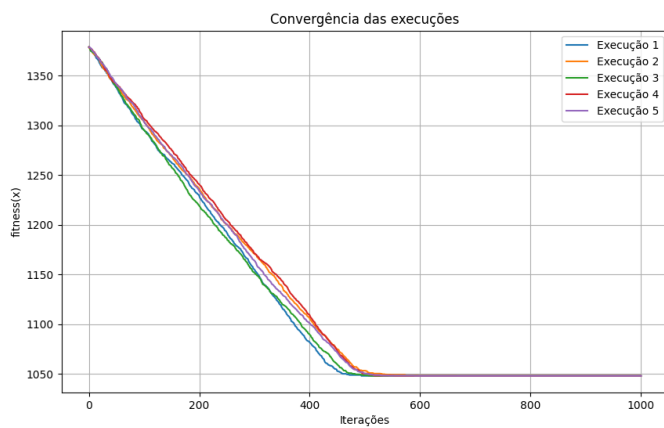


Fig. 3. Convergência do BVNS implementado para o Problema 2 Isolado.

B. Abordagem de Soma Ponderada

Para a soma ponderada, usamos a modelagem de (9) com 20 valores aleatórios de w entre 0 e 1, executando 5 vezes, obtemos a Fronteira Pareto-ótima exibida na Figura 4. A fronteira exibe os valores absolutos das funções objetivo, mas elas foram obtidas considerando-se a função objetivo normalizada por (13).

Ao analisarmos os resultados para as 5 execuções foi notado que quando w assumia valores maiores que aproximadamente 0.65 a solução converge para o valor ótimo de f_1 e quando assumia valores menores que 0.15 a solução converge para o valor ótimo de f_2 . Ou seja, há uma perda de resolução nos pontos da fronteira devido a essa sensibilidade aos pesos escolhidos aleatoriamente.

C. Abordagem ϵ -Restrito

Para a abordagem ϵ -Restrito, geramos 20 valores de ϵ_2 espaçados igualmente no intervalo $[\min f_2, \max f_2]$, obtendo a fronteira Pareto-ótima exibida na Figura 5.

Comparando a fronteira do ϵ -Restrito com a da soma ponderada, vemos que há uma menor sobreposição entre

Fig. 4. Fronteira Pareto-ótima obtida via BVNS com soma ponderada.

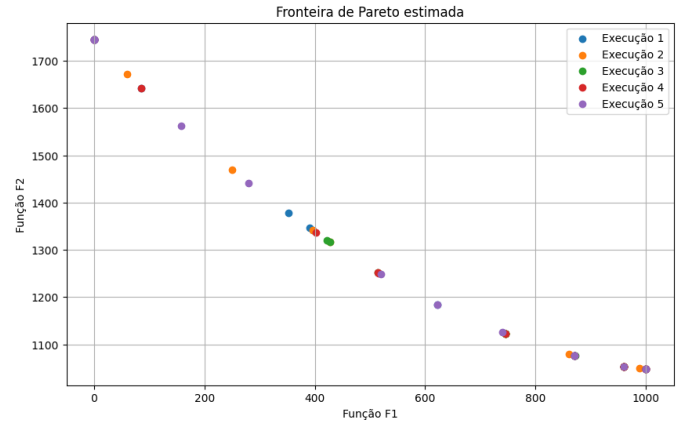
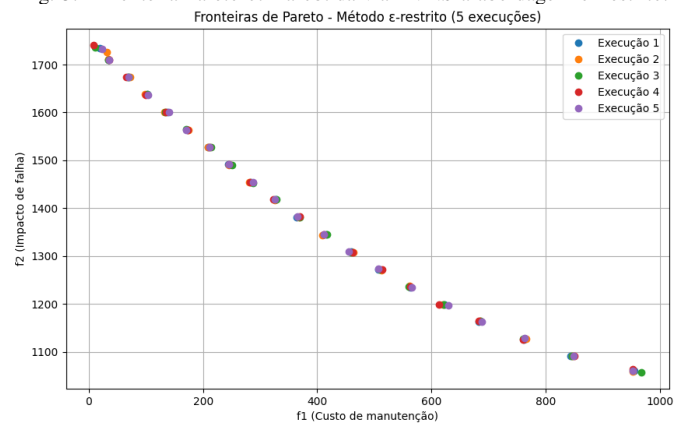


Fig. 5. Fronteira Pareto-ótima obtida via BVNS a abordagem ϵ -Restrito.



os pontos da fronteira. Obtendo assim um maior número de soluções distintas. Dessa forma o método ϵ -Restrito foi escolhido para a geração da Fronteira Pareto que será tomada a decisão problema multiobjetivo.

D. Indicador de Hipervolume (HVI)

Para a realização do cálculo a métrica de HVI foi permitido o uso de mais de 20 soluções pareto na fronteira, de maneira que o valor para a métrica atingisse os valores indicados para comparação. Dessa forma, foram utilizadas 200 soluções e a visualização da fronteira pode ser vista na Figura 6.

O valor de HVI obtido foi:

$$\text{HVI: } 0.602974 \quad (19)$$

E. Tomada de Decisão Multicritério

Para a tomada de decisão multicritério, vamos considerar 20 dentre as 200 soluções exibidas na Figura 6. Vamos comparar as soluções usando 4 atributos de interesse, definidos abaixo. A justificativa dos atributos 3 e 4 será dada a seguir.

- 1) Valor de f_1 ;
- 2) Valor de f_2 ;
- 3) Razão entre o número de equipamentos no plano 1 pelo número no plano 3;

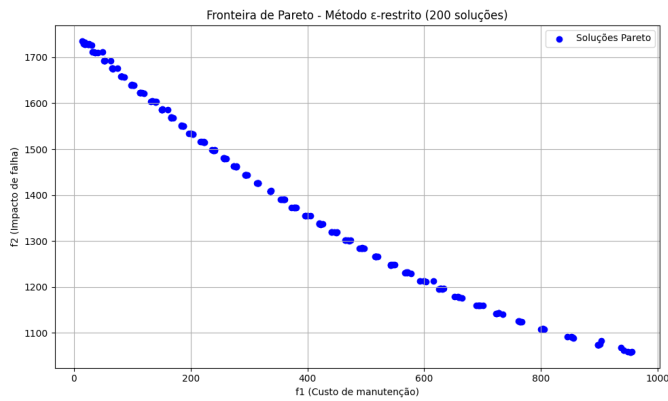


Fig. 6. Fronteira Pareto para 200 soluções com método ϵ -restrito usada para cálculo do HVI

- 4) Soma do número de equipamentos do Cluster 4 no plano 1 e o número de equipamentos de equipamentos do Cluster 1 no plano 1.

Para entender o atributo 3, vamos analisar algumas soluções da fronteira em termos dos equipamentos, como mostra a Figura 7. Quando o ϵ é pequeno, estamos priorizando a função f_2 e assim o algoritmo escolhe a solução que coloca quase todos no plano mais caro (M3), a fim de minimizar o custo esperado de falha. Quando ϵ é pequeno, priorizamos f_1 e o algoritmo coloca a manutenção mais barata (M1) para todos, minimizando o custo total. Quando ϵ está no meio termo, estamos no meio da fronteira Pareto e ele escolhe uma solução "meio-termo".

É importante notar que o plano M2 na Figura 7 quase nunca é utilizado. Assim, podemos analisar apenas as distribuições entre M1 e M3 para ter uma noção de como os equipamentos estão sendo distribuídos entre os planos. Além disso, podemos considerar que é indesejável colocar todos os equipamentos em apenas um plano: se a empresa tem 3 planos de manutenção, é esperado que os equipamentos sejam distribuídos ao longo desses três planos, evitando-se colocar todos os equipamentos em apenas um.

Assim, definimos o atributo 3: a razão entre o número de equipamentos no plano 1 pelo número no plano 3. Quanto mais próximo de 1 esse número estiver, melhor a classificação da solução, indicando que os equipamentos estão igualmente distribuídos ao longo de M1 e M3.

Agora vamos analisar como os valores de f_2 se comportam para diferentes clusters, a fim de entender melhor o que eles significam. A Figura 8 mostra o valor de f_2 para diferentes clusters. Vemos que f_2 - o custo esperado de falha - é menor para os equipamentos do Cluster 4 do que para os do Cluster 1. Esse padrão se repete para os três planos de manutenção. Logo, podemos interpretar isso como um indicativo de que os equipamentos do Cluster 1 são mais valiosos do que os equipamentos do Cluster 4, uma vez que o custo de falha deles é maior. Assim, quanto mais equipamentos do cluster 1 estiverem no plano mais caro (M3) e quanto mais equipamentos

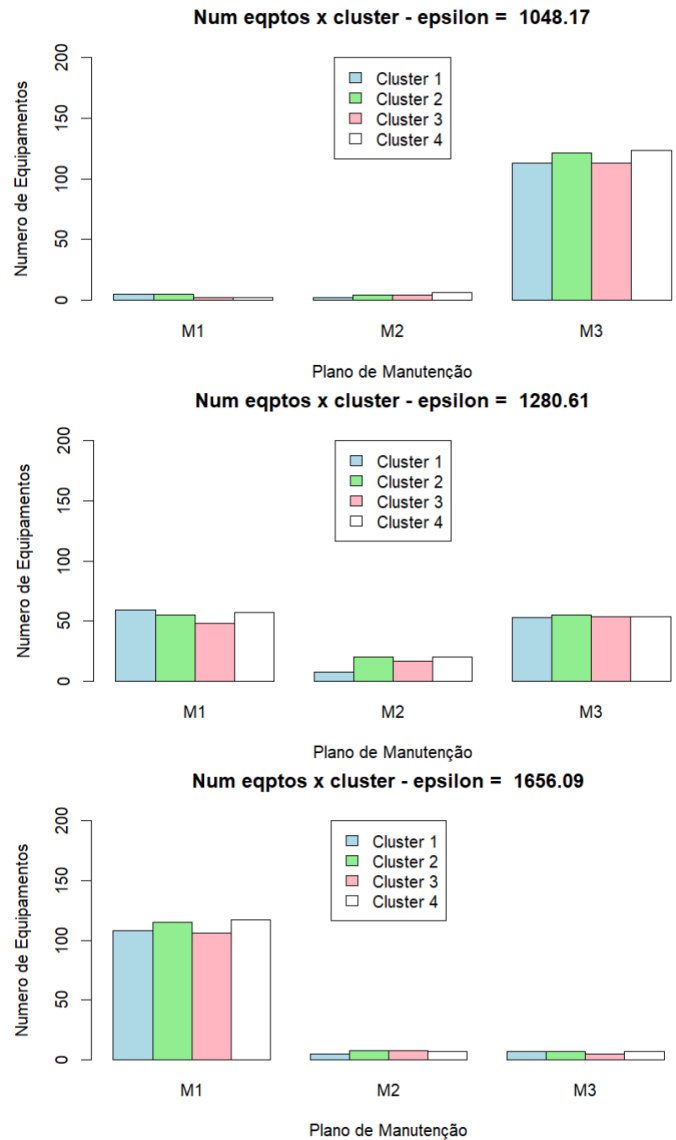


Fig. 7. Distribuição de equipamentos por plano de manutenção para diferentes valores de ϵ .

do cluster 4 estiverem no mais barato (M1), melhor.

Dessa forma, definimos o atributo 4: soma do número de equipamentos do Cluster 4 no plano 1 e o número de equipamentos de equipamentos do Cluster 1 no plano 1. Quanto maior esse número, melhor.

Para as 20 soluções escolhidas aleatoriamente, calculamos os valores dos 4 atributos, obtendo os resultados da Figura 9.

F. Solução via AHP

Usando o AHP, temos que montar as tabelas de comparação entre as 20 soluções para cada atributo, e depois uma tabela de comparação entre os atributos em si. Como os atributos são todos numéricos, elas podem ser feitas programaticamente através de condicionais. Por exemplo, para o atributo 3, se uma solução possui A3 igual a 1.2 e outra possui A3 = 20,

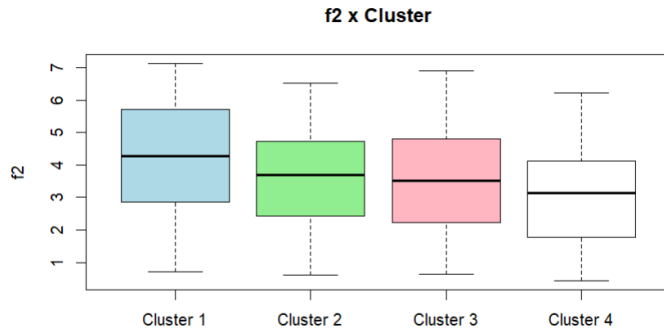


Fig. 8. Valor de f_2 para diferentes clusters.

Solução	f1	f2	A3	A4
1	765	1,124	0.27	121
2	36	1,710	78.33	120
3	728	1,143	0.31	115
4	14	1,736	100.00	125
5	450	1,320	1.26	110
6	420	1,339	1.45	118
7	517	1,266	0.93	110
8	763	1,126	0.26	117
9	956	1,059	0.03	115
10	397	1,356	1.61	108
11	80	1,658	17.15	124
12	569	1,232	0.73	112
13	603	1,211	0.63	121
14	86	1,657	25.35	116
15	466	1,302	1.17	114
16	297	1,444	2.72	116
17	65	1,676	32.07	115
18	114	1,623	12.70	117
19	497	1,284	1.01	110
20	261	1,480	3.46	111

Fig. 9. Valores calculados para os 4 atributos para as 20 soluções consideradas.

colocamos o valor 9 na tabela indicando que a primeira é bem melhor do que a segunda.

O vetor de prioridades de cada tabela é calculado através do autovetor direito principal, e o índice de inconsistência (IC) é calculado para cada um das tabelas. A tabela de prioridades dos atributos tem dimensão 4 x 4 e está exibida na Figura 10.

Att	f1	f2	A3	A4
f1	1.00	0.20	5	3.00
f2	5.00	1.00	5	3.00
A3	0.20	0.20	1	0.33
A4	0.33	0.33	3	1.00

Fig. 10. Tabela de atributos do AHP. IC = 0.12.

Na Figura 10, colocamos alta prioridade para o atributo f_1 . Nesse caso, a solução encontrada pelo AHP está indicada na Figura 11. Ele selecionou a solução que coloca todos no mais barato, de fato priorizando f_1 , conforme esperado.

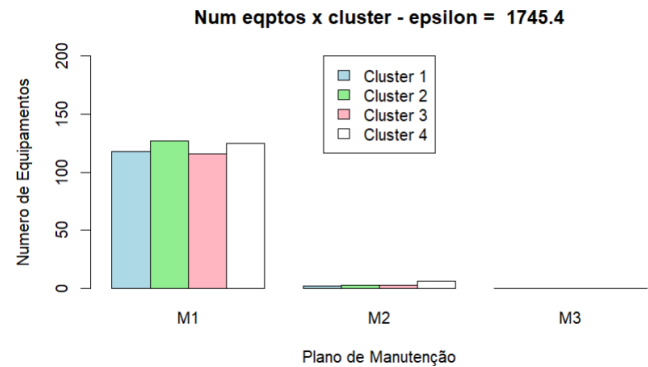


Fig. 11. Tabela de atributos do AHP. IC = 0.12.

Modificando as prioridades na tabela de atributos, a solução selecionada muda para atender às novas relações entre os atributos. Por exemplo, colocando alta prioridade para o atributo A3 em relação aos demais, temos a solução indicada na Figura 12. Ele seleciona a solução em que a razão entre M1 e M3 é próxima de 1, conforme definimos o atributo.

As localizações na fronteira Pareto das soluções nas Figuras 11 e 12 estão exibidas na Figura 13.

G. Solução via PROMETHEE

Análise dos resultados obtidos a partir da aplicação do método PROMETHEE II sobre um conjunto de soluções não-dominadas, considerando os 4 atributos definidos acima.

Foram testadas três configurações distintas de pesos atribuídos a esses critérios, priorizando cada um deles individualmente.

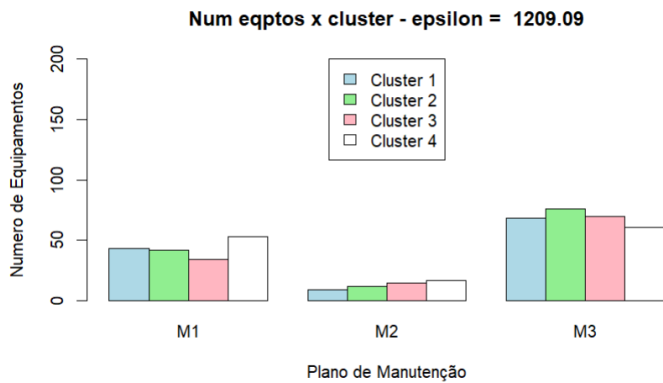


Fig. 12. Tabela de atributos do AHP. IC = 0.12.

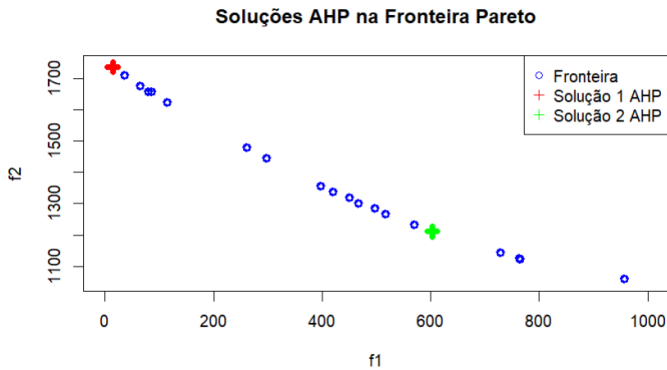


Fig. 13. Localizações das soluções encontradas pelo AHP na Fronteira Pareto.

1) Prioridade Alta para f_1 (Custo de Manutenção):

- Pesos utilizados: $f_1 = 0.50$, $f_2 = 0.20$, $a_3 = 0.15$, $a_4 = 0.15$
- Índice: 6, $f_1 = 170.00$, $f_2 = 1568.02$, $a_3 = 7.47$, $a_4 = 108$

A Figura 14 mostra a solução encontrada para o Promethee priorizando-se o valor de f_1 .

Nesta configuração, a solução selecionada apresenta o menor custo de manutenção entre todas as avaliadas, o que é coerente com o peso elevado atribuído ao critério f_1 . No entanto, o custo de falha (f_2) é consideravelmente mais alto, indicando que o baixo investimento em manutenção aumenta os riscos de falhas. A razão entre M1 e M3 (a_3) é bastante alta, reforçando que muitos equipamentos foram designados ao plano M1.

2) Prioridade Alta para f_2 (Custo Esperados de Falha):

- Pesos utilizados: $f_1 = 0.10$, $f_2 = 0.60$, $a_3 = 0.15$, $a_4 = 0.15$
- Índice: 9, $f_1 = 734.00$, $f_2 = 1140.97$, $a_3 = 0.30$, $a_4 = 111$

A Figura 15 mostra a solução encontrada para o Promethee priorizando-se o valor de f_2 .

A configuração priorizou a minimização do custo esperado de falha, e a solução escolhida apresenta um f_2 significativamente mais baixo. O custo de manutenção é muito elevado, o que indica uma alocação mais intensiva em planos preventivos mais

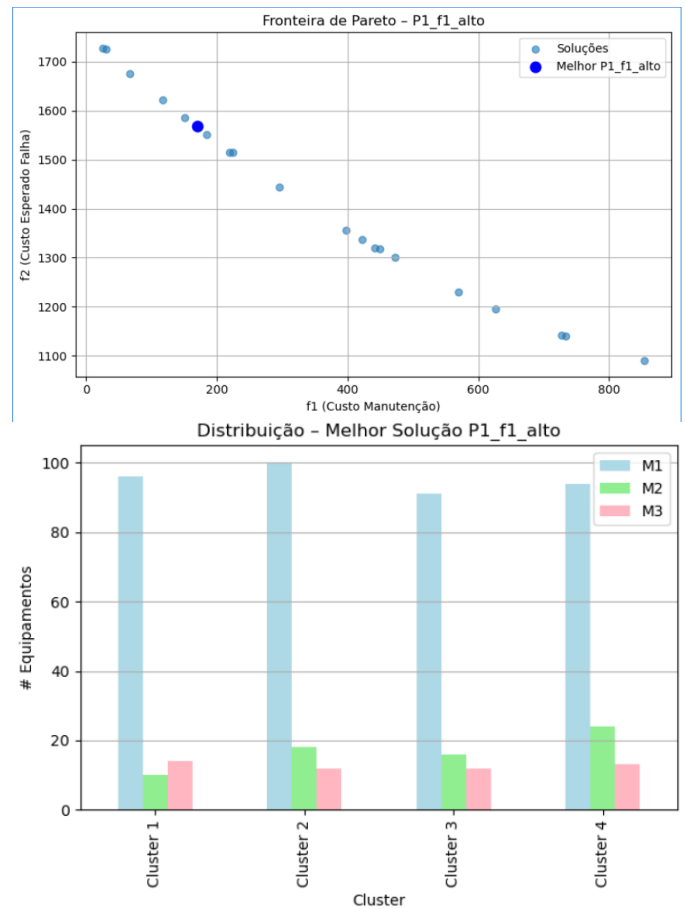


Fig. 14. Solução encontrada para o Promethee priorizando-se o valor de f_1 .

custosos. A razão a_3 é baixa, sugerindo uma maior proporção de equipamentos em M3.

3) Prioridade Alta para A_3 (Razão entre M1 e M3):

- Pesos utilizados: $f_1 = 0.30$, $f_2 = 0.30$, $a_3 = 0.30$, $a_4 = 0.10$
- Índice: 11, $f_1 = 295.00$, $f_2 = 1444.02$, $a_3 = 2.69$, $a_4 = 107$

A Figura 16 mostra a solução encontrada para o Promethee priorizando-se o valor de a_3 .

Com foco na razão entre planos M1 e M3, a solução resultante apresenta um equilíbrio moderado entre custo de manutenção e risco de falha. A razão a_3 está próxima de 3, o que indica uma leve preferência por M1 em relação a M3. A solução mantém também valores intermediários para f_1 e f_2 , sendo uma alternativa de compromisso.

4) Considerações Finais: A análise evidencia que diferentes configurações de pesos nos critérios de decisão impactam diretamente o perfil da solução escolhida. Soluções com menor custo de manutenção tendem a apresentar maior risco de falhas, enquanto aquelas que minimizam falhas demandam maior investimento em manutenção. O critério a_3 mostrou ser sensível à distribuição entre planos extremos (M1 e M3), enquanto o critério a_4 contribui para favorecer a cobertura estratégica de

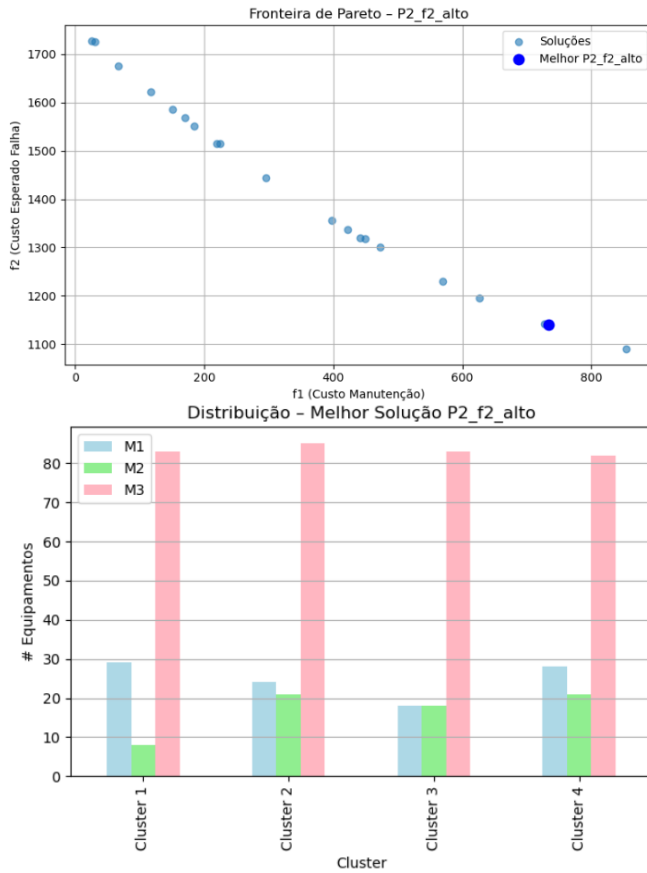


Fig. 15. Solução encontrada para o Promethee priorizando-se o valor de f_2 .

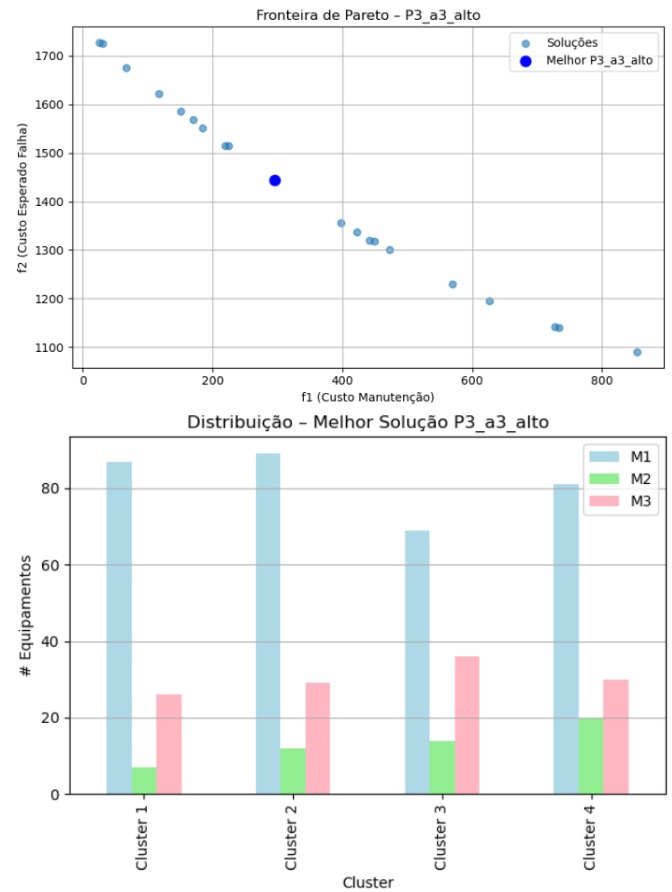


Fig. 16. Solução encontrada para o Promethee priorizando-se o valor de f_2 .

certos clusters.

IV. CONCLUSÃO

O algoritmo implementado para o BVNS e suas estruturas de vizinhança e estratégias de refinamento executaram bem o papel esperado. Encontrando de forma satisfatória os resultados ótimos individuais das funções (f_1) e (f_2).

As duas abordagens clássicas comparadas para a geração da fronteira pareto evidenciou que a abordagem ϵ -Restrito forneceu uma fronteira mais diversificada, com menor sobreposição entre soluções, sendo mais adequada para a aplicação de métodos de tomada de decisão seguintes.

Avaliando-se 200 soluções não-dominadas por meio do indicador de Hipervolume (HVI), foi obtido um valor de $HVI = 0.602974$, indicando boa qualidade e diversidade na fronteira aproximada.

Na etapa de tomada de decisão multicritério, foram definidos quatro atributos com base em análises estruturadas sobre os dados do problema e o comportamento das soluções. Os métodos clássicos de apoio à decisão AHP e PROMETHEE foram utilizados para selecionar as soluções mais adequadas conforme diferentes perfis de preferência, destacando os ganhos e perdas entre os critérios envolvidos e a valorização do atributo A_3 para uma solução ótima mais neutra.

Os resultados obtidos reforçam a importância da integração entre técnicas de otimização e métodos multicritério na escolha de soluções em problemas complexos. Além disso, a abordagem adotada mostra-se flexível e eficaz, podendo ser adaptada a outros cenários de decisão com múltiplos objetivos e restrições.

REFERÊNCIAS

- [1] Bode, Felix & Reed, Patrick & Reuschen, Sebastian & Nowak, Wolfgang. (2019). Search Space Representation and Reduction Methods to Enhance Multi-Objective Water Supply Monitoring Design. Water Resources Research. 55. 10.1029/2018WR023133.