

Un método de generación aleatoria de pesos enteros ordenados aplicado al análisis post Pareto del problema de asignación de componentes redundantes en un sistema complejo

Víctor M. Carrillo¹, Emmanuel Gurrola²

¹Universidad Autónoma de Ciudad Juárez.

²The University of Texas at El Paso

Resumen

Este trabajo propone una nueva alternativa para la generación numérica de pesos ordenados para ayudar en la reducción del tamaño del conjunto de Pareto en el problema de asignación de componentes redundantes (ACR) en un sistema complejo con tres o más funciones objetivo. Este nuevo método genera una sucesión finita creciente de números aleatorios enteros positivos utilizados para construir un conjunto de pesos ordenados y clasificados para una función que es una combinación lineal de las funciones objetivo del problema de optimización multiobjetivo que resulta en la asignación de componentes redundantes(ACR). Se presentan dos ejemplos ilustrativos para mostrar y comparar el rendimiento del método con otros generadores de pesos ordenados ya conocidos.

Palabras clave: optimización multi-objetivo, conjunto Pareto-óptimo, algoritmo de generación de pesos ordenados enteros. Problema de Asignación de componentes redundantes (PACR).

Introducción

Dentro de los problemas de optimización multi-objetivo (MOP), no es raro ver que las funciones objetivo están en conflicto entre ellas. El problema de la asignación de componentes redundantes en un sistema para incrementar su confiabilidad (RAP) no es una excepción. Un enfoque reciente para resolver MOPs implica generar el conjunto Pareto solución óptima y luego la combinación de todos los objetivos en una sola función. Los pesos se asignan a cada objetivo para reflejar la preferencia quien

toma decisiones. Sin embargo, comúnmente se hace engorroso en la práctica real. En este trabajo se propone un método para obtener cualquier número de pesos para la construcción de una función lineal para cualquier la reducción del conjunto optimo obtenido por cualquiera de los métodos arriba mencionados al resolver un problema de optimización multiobjetivo. En este trabajo se aplica la generación de pesos para la reducción de un conjunto de soluciones

Revisión de Literatura

Conjuntos de Pareto

En los últimos años se han desarrollado y utilizado varios métodos meta heurísticos para generar el conjunto de soluciones de Pareto y resolver el RAP como los algoritmos genéticos, la colonia de hormigas, enjambre de partículas, etc. Los algoritmos genéticos por ejemplo han atraído a los investigadores debido a varios resultados exitosos reportados para obtener conjuntos de Pareto (Ando y Suzuki, 2006).

El problema de asignación de componentes redundantes (RAP) es un ejemplo común de un MOP encontrado en la literatura. Los intentos anteriores para resolver el RAP involucrados tratar el problema como un problema de una sola función objetivo (single objective problem- SOP) donde la confiabilidad del sistema es el objetivo a maximizar. La minimización de costos y otras funciones objetivo son tratadas como restricciones del problema. En Wang et al. (2009), el RAP se formuló como un problema de asignación combinatoria de componentes redundantes de sistemas en serie-paralelo. Un algoritmo multiobjetivo evolutivo (MOEA) conocido como Non-dominated Sorting Genetic Algorithm II (NSGA-II) fue empleado para resolver los problemas de asignación de componentes redundantes en problemas multi-objetivo. El experimento demostró que un enfoque con MOEAs puede proporcionar soluciones satisfactorias.

En el artículo de Taboada et al. (2008) fue presentado un algoritmo genético para resolver problemas multi objetivo de confiabilidad multi-estado (MOMS). Se usaron dos ejemplos para mostrar el potencial del algoritmo. El ejemplo1 consistió en 5 unidades conectadas en serie y el ejemplo2 de 3 unidades también conectados en serie. Cada unidad principal contenía varios componentes las cuales están disponibles en el mercado. El problema consistía en maximizar la disponibilidad del sistema, reduciendo al mínimo el costo y el peso del sistema. Con un tamaño de población de 200 y 50 generaciones, el algoritmo MOMS-GA genero un frente de Pareto que contiene 118 soluciones en un tiempo de 595.25 segundos. En el ejemplo 2 con un tamaño de población de 100 y 50 generaciones el mismo algoritmo produjo un conjunto de Pareto de 57 soluciones no dominadas en 606.20 segundos.

Kumar et al. propusieron una formulación de objetivos múltiples, así como un método para resolver el problema de optimización de asignación de la redundancia multinivel (MRAOP). Este problema es comúnmente encontrado en muchos sistemas complejos en ingeniería los cuales se conocen como sistemas jerárquicos. La metodología de genotipo jerárquica se constituyó combinando el algoritmo NSGA-II y la fortaleza del algoritmo genético evolutivo SPEA2 para la obtención del frente de Pareto. Los métodos de optimización

multiobjetivo permite a los tomadores de decisiones evaluar las soluciones obtenidas (Coit, Jin, y Tekiner, 2009). Sin embargo, el gran número de soluciones de Pareto generadas puede convertirse en otro problema para elegir la mejor solución. Taboada et al. (2006) propusieron una metodología práctica para la solución del problema de optimización de la confiabilidad de un sistema multi-objetivo consistente en una técnica de validación de clusters al interior del algoritmo de las k-means. Con este enfoque, el decisor obtiene un subconjunto de Pareto reducido de simplemente k soluciones particulares. Este enfoque de agrupación de datos se utilizó para analizar un RAP Multi-objetivo.

Una propuesta reciente para reducir el tamaño de Pareto fue desarrollada por Li et al. (2009). Un conjunto de soluciones óptimas de Pareto se identificó inicialmente en la primera etapa al aplicar un algoritmo evolutivo multiobjetivo (MOEA), en la segunda etapa se aplicó un mapa auto organizado (SOM) para clasificar las soluciones óptimas de Pareto y agruparlas en varios clusters. Posteriormente se realizó, el análisis envolvente de datos (DEA) mediante la comparación de la eficiencia relativa de esas soluciones para determinar las soluciones finales representativas para el problema general. En Hui et al. un algoritmo multi-objetivo multi-estado diferencial (MOMS-HDEA) se describe. El algoritmo se aplica para resolver el problema de la disponibilidad del sistema multi-estado multi-objetivo.

Reducción del tamaño del conjunto de Pareto

El algoritmo para la reducción del conjunto de Pareto se basa en una combinación de los dos enfoques generales descritos en Konak et al. para resolver problemas de selección múltiple objetivo (MOP). El primer paso consiste en encontrar la solución de conjunto Pareto con uno de los métodos descritos anteriormente. Una vez que el conjunto óptimo de Pareto se conoce todos los objetivos se combinan en una sola función. Los métodos que se encuentran en esta categoría incluyen teoría de la utilidad y el método de suma ponderada. (Istanbul, Coit y Smith).

Este trabajo se centra en el uso del método de suma ponderada como una herramienta para realizar el análisis post Pareto. El algoritmo está inspirado en las obras analizadas en Taboada et al. (2006) y Carrillo et al. (2011). Sin embargo, su desarrollo se basa en los números enteros pseudo aleatorios y no en funciones de incertidumbre. El MOP se analiza en Taboada et al. y se presenta de la siguiente manera:

Minimizar

$$\begin{aligned} f(x) &= [f_1(x), \dots, f_k(x)] \\ s. t. \quad g_j(x) &\leq 0 ; j = 1, \dots, m \end{aligned} \quad (1)$$

Dónde:

n es el número de funciones objetivo,

m es el número de restricciones de desigualdad,

$x \in \mathfrak{R}^m$ es el vector de variables del diseño,
y

$f(x) \in \mathfrak{R}^k$ es el vector de funciones objetivo.

$$f_i: \mathfrak{R}^n \rightarrow \mathfrak{R} \text{ para } i = 1, \dots, k$$

La región factible se define como $X : x \text{ tal que } g_j(x) \leq 0 \text{ } |; j = 1, \dots, m$. La imagen de la región factible es $Z = \{f(x) | x \in X\}$. La obtención de la solución de la ecuación (1) puede no obtenerse al no existir un punto que minimice todas las funciones objetivo simultáneamente. Por lo anterior el concepto de optimalidad de Pareto se usa para describir las soluciones

óptimas de un problema de optimización multiobjetivo. Una solución se dice óptima de Pareto si no es posible moverse desde dicho punto a otro para mejorar al menos una función objetivo en detrimento de otra función objetivo. Alternativamente un punto se dice que es un punto débil de Pareto si no es posible moverse desde ese punto y mejorar todas las funciones objetivo simultáneamente.

Como se mencionó anteriormente la suma ponderada puede ser expresada como la función de utilidad.

$$U = w_1 f_1(x) + \dots + w_k f_k(x) \text{ función de utilidad} \quad (2)$$

Tal que $w_1 + w_2 + \dots + w_k = 1$ y $0 \leq w_i \leq 1 \forall k$. Si todos los pesos w_i son positivos, como se supone en este trabajo, entonces se satisfacen las condiciones

suficientes de optimalidad, lo que significa que el mínimo de la ecuación (2) es un punto óptimo de Pareto (Marler and Arora, 2008).

Metodos

Rankeo no numérico para tres pesos ponderados

Este es un método de optimalidad post Pareto propuesto y desarrollado por Taboada et al. (2006). Inicialmente las funciones objetivo son dispuestas en orden de importancia por el decisor como se muestra $f_1 > f_2 > \dots > f_n \Rightarrow w_1 > w_2 > \dots > w_n$ y después los pesos se generan de acuerdo a la ordenación de las funciones objetivo. Posteriormente una función de densidad de

probabilidad $f_w(w)$ es desarrollada basada en el orden deseado de los pesos ponderados $w = (w_1, w_2, \dots, w_n)$ donde se debe satisfacer la condición $w_1 + w_2 + \dots + w_n = 1$ como abajo se muestra para el caso de tres funciones objetivo. En efecto, en la ecuación 3 se muestra la función de densidad de probabilidad f.d.p. la cual proveerá los pesos ordenados w_1 para la construcción de la función de utilidad en la ecuación (2)

$$f_w(w_1, w_2, w_3) = \begin{cases} c, & w_1 > w_2 > w_3 \\ 0, & \text{otro caso} \end{cases} \text{ donde } w_1 + w_2 + w_3 = 1 \text{ y } 0 \leq w_i \leq 1 \quad (3)$$

La f.d.p (3) se reduce a una función de dos dimensiones Ec. 4

$$f_w(w_1, w_2) = \begin{cases} c, & w_1 > w_2 > 1 - w_1 - w_2 \\ 0, & \text{de otra manera} \end{cases} \quad (4)$$

$$f_w(w_1, w_2) = \begin{cases} 0 & 0 \leq w_1 \leq \frac{1}{3} \\ 12 & \frac{1}{2} - \frac{1}{2}w_1 \leq w_2 \leq w_1 & \& \frac{1}{3} \leq w_1 < \frac{1}{2} \\ 12 & \frac{1}{2} - \frac{1}{2}w_1 \leq w_2 \leq 1 - w_1 & \& \frac{1}{2} \leq w_1 < 1 \end{cases} \quad (5)$$

Luego su función de distribución acumulada (f.d.a.) es Ec. 6:

$$F_{w_1}(w_1) = \int_{-\infty}^{w_1} f_{w_1}(w_1)dw_1 = \begin{cases} 0 & , & 0 \leq w_1 < \frac{1}{3} \\ 9w_1^2 - 6w_1 + 1 & , & \frac{1}{3} \leq w_1 < \frac{1}{2} \\ -3w_1^2 + 6w_1 - 2 & , & \frac{1}{2} \leq w_1 < 1 \end{cases} \quad (6)$$

De la cual al usar el teorema integral de probabilidad se despeja w_1 Ec. 7

$$w_1 = F_{w_1}^{-1}(u) = \begin{cases} 0 & , \text{otro caso} \\ \frac{\sqrt{u+1}}{3} & , 0 \leq u < \frac{1}{4} \\ 1 - \sqrt{\frac{1-u}{3}} & , \frac{1}{4} \leq u < 1 \end{cases} \quad (7)$$

De forma similar se calcula w_2 Ec. 8

$$w_2 = F_{w_2}^{-1}(u|w_1) = \begin{cases} 0 & , & 0 \leq w_1 < \frac{1}{3} \\ \frac{(3w_1 - 1)u + 1 - w_1}{2} & , 0 \leq u \leq 1 & \frac{1}{3} \leq w_1 < \frac{1}{2} \\ \frac{(1 - w_1)(u + 1)}{2} & , 0 \leq u \leq 1 & \frac{1}{2} \leq w_1 < 1 \end{cases} \quad (8)$$

Para finalizar el proceso se despeja w_3 de la ecuación $w_1 + w_2 + \dots + w_n = 1$. Una vez que los pesos son obtenidos se genera un conjunto de combinaciones lineales o valores de la función de utilidad $U = w_1f_1(x) + \dots + w_kf_k(x)$ de los cuales se seleccionan aquellas combinaciones que

tengan mayor frecuencia relativa para elegir de estos los mejores puntos óptimos de Pareto.

Algoritmo generador de pesos enteros ordenados (AGPEO)

En esta sección se presenta el pseudocódigo del algoritmo a desarrollar así como su

desarrollo matemático.

Pseudocódigo del algoritmo

Algoritmo generador de pesos enteros
1: Inicio del algoritmo
2: Sea n el número de pesos a ser generados
3: Sea K un número mayor a la suma de n enteros
4: Calcule cotas superiores e inferiores para el primer peso UB_1 y LB_1
5: Generar un entero aleatorio w_1 entre UB_1 y LB_1
6: for $i = 1$ hasta $n - 1$
7: dado w_{i-1} calcular UB_i y LB_i
8: generar un entero aleatorio w_i entre UB_i y LB_i
9: end for
10: El ultimo valor se obtiene igualando $w_n = K - w_1 + w_2 + \dots + w_{n-1}$
11: Fin del algoritmo

Figura 1. Pseudocódigo del generador de pesos enteros

Desarrollo del algoritmo

Como se mostró en la figura 1, el algoritmo genera una sucesión finita ordenada de enteros positivos w_1 donde $w_1 > w_2 > \dots > w_n > 0$ para un entero positivo K ; que satisfaga $K - [(n(n-1)) * (2)^{-1}] > 0$. Para generar el primer valor w_1 se calculan las cotas superior e inferior.

$$\text{Cota superior: } w_{1max} = K - \left\lfloor \frac{n(n-1)}{2} \right\rfloor \quad (9)$$

$$\alpha_i = K - \sum_{j=1}^{i-1} w_j - \left\lfloor \frac{(n-i+1)(n-i)}{2} \right\rfloor \quad i = 2, \dots, n-1 \quad (11)$$

$$\text{Si } \alpha_i \leq w_{i-1} \text{ entonces } w_{imax} = \alpha_i \text{ en otro caso } w_{imax} = w_{i-1} - 1 \quad (12)$$

$$\text{Cota inferior: } w_{1min} = \left\lceil \frac{K + \left\lfloor \frac{n(n-1)}{2} \right\rfloor}{n} \right\rceil \quad (10)$$

Entonces w_1 es un número entero aleatorio entre las cotas w_{1min} and w_{1max} .

Para obtener los pesos posteriores se calculan las cotas superiores e inferiores de acuerdo a la siguiente regla:

Para determinar la cota superior de w_1 calcular

Para determinar la cota inferior de w_i use la ecuación (13)

$$w_{imin} = \left\lceil \frac{K - \sum_{j=1}^{i-1} w_j + \left\lceil \frac{(n-i+1)(n-i)}{2} \right\rceil}{n-i+1} \right\rceil \quad (13)$$

Finalmente w_i es igual a un número entero aleatorio entre w_{imin} y w_{imax} donde $i = 2, \dots, n-1$.

El último peso se obtiene por:

$$w_n = K - \sum_{j=1}^{n-1} w_j \quad (14)$$

Una vez que los n enteros son generados se calcula una sucesión ordenada de números entre 0 y

1 al dividir $w_i^* = w_i \cdot K^{-1}$ para $i = 1, \dots, n$ donde $w_1 > w_2 > \dots > w_n > 0$ y $w_1 + w_2 + \dots + w_n = K$.

Resultados

Ejemplo numérico

Para mostrar las potenciales aplicaciones del algoritmo generador de pesos enteros, se usó un valor de $K = 1000$ para reducir el tamaño de un conjunto de Pareto consistente de 75 puntos presentado por Taboada et al.

(2006), como se muestra en la tabla 1 y la figura 2. Se asignaron distintos valores de importancia a las ponderaciones de la confiabilidad, costo y peso para generar varios escenarios que un decisor pudiera enfrentar.

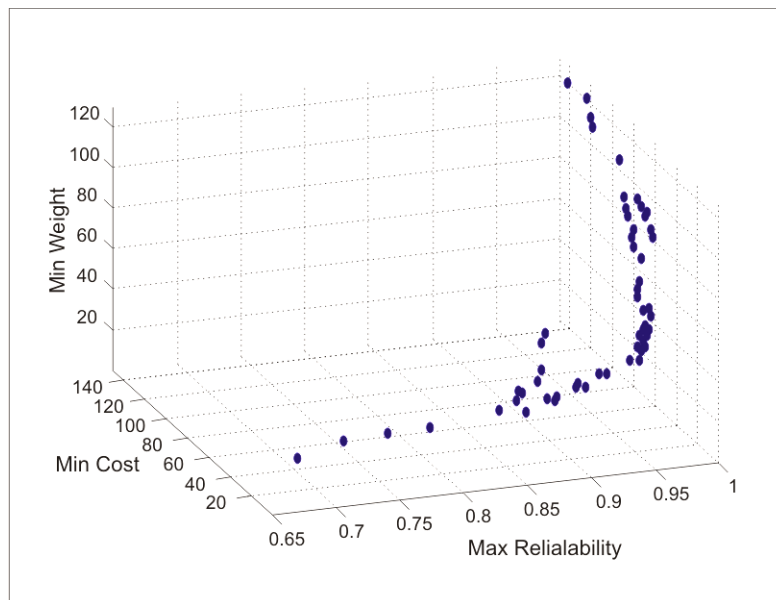


Figura 2. Conjunto solución de Pareto

Tabla 1. Conjunto solución de Pareto, Max Confiabilidad, Min Costo & Min Peso

Sol. #	Reliability	Cost	Weight	Sol. #	Reliability	Cost	Weight	Sol. #	Reliability	Cost	Weight
1	0.68204 8	13	19	26	0.96188 3	38	36	51	0.99804 3	68	45
2	0.72036 5	16	24	27	0.96227 2	39	34	52	0.99805 9	69	43
3	0.75707 3	19	23	28	0.96364 4	39	35	53	0.99811 9	73	55
4	0.78860 1	17	25	29	0.96842 5	40	40	54	0.99844 3	75	47
5	0.84319 7	19	28	30	0.97019 8	41	39	55	0.99883 6	72	67
6	0.85957 5	21	31	31	0.97303 5	42	34	56	0.99903 6	77	51
7	0.860364	21	36	32	0.973429	43	32	57	0.999049	79	69
8	0.864489	23	34	33	0.974604	42	41	58	0.999054	80	69
9	0.875328	31	20	34	0.979653	44	38	59	0.999074	82	73
10	0.877076	23	39	35	0.980835	43	47	60	0.999363	62	82
11	0.88209	25	42	36	0.982178	52	35	61	0.999565	81	77
12	0.883349	27	55	37	0.982441	45	45	62	0.999632	64	85
13	0.888398	29	58	38	0.982575	53	33	63	0.999721	70	89
14	0.893196	33	24	39	0.982762	54	39	64	0.999732	74	91
15	0.901223	34	24	40	0.983483	46	46	65	0.999783	68	92
16	0.901588	35	22	41	0.984387	50	42	66	0.999813	76	94
17	0.915556	33	28	42	0.985093	48	44	67	0.999868	85	81
18	0.917232	34	27	43	0.986017	54	43	68	0.999937	87	85
19	0.919619	36	28	44	0.986416	55	41	69	0.999942	88	85
20	0.926172	37	26	45	0.990477	56	37	70	0.999962	90	89
21	0.934245	35	32	46	0.991066	58	41	71	0.999977	94	105
22	0.935955	36	31	47	0.992115	59	37	72	0.999982	119	110
23	0.942641	36	32	48	0.992262	57	50	73	0.999994	121	113
24	0.944749	38	29	49	0.994349	58	45	74	0.999998	125	121
25	0.945077	39	30	50	0.99494	60	49	75	0.999999	143	120

Tabla 2. AGPE Posibles Escenarios

A	Confiabilidad > Costo > Peso
B	Confiabilidad > Peso > Costo
C	Costo > Confiabilidad > Peso
D	Peso > Confiabilidad > Costo

Escenarios

El número de soluciones obtenidas con el algoritmo AGPE para $K = 1,000$ no se pueden distinguir a los obtenidos con el método no-numérico ranking NNR. Para el Escenario A, el algoritmo AGPE proporciona 19 soluciones en comparación con 17 a partir del método NNR. Las figuras 3 a 10 y la Tabla 3 muestran las soluciones resultantes para el Escenario A. En el

Escenario B el método NNR proporcionó más soluciones, 10 vs. 9. Los resultados para el Escenario B se muestran en las figuras 11-18 y en la Tabla 4. En el Escenario C del NNR también generó una solución adicional, 13 vs 12, como se muestra en las figuras 19-26 y en la Tabla 5. Para el escenario D ambos métodos coinciden en el número de soluciones: cinco soluciones. Las figuras 27-34 y Tabla 6 muestran las soluciones obtenidas para el Escenario D.

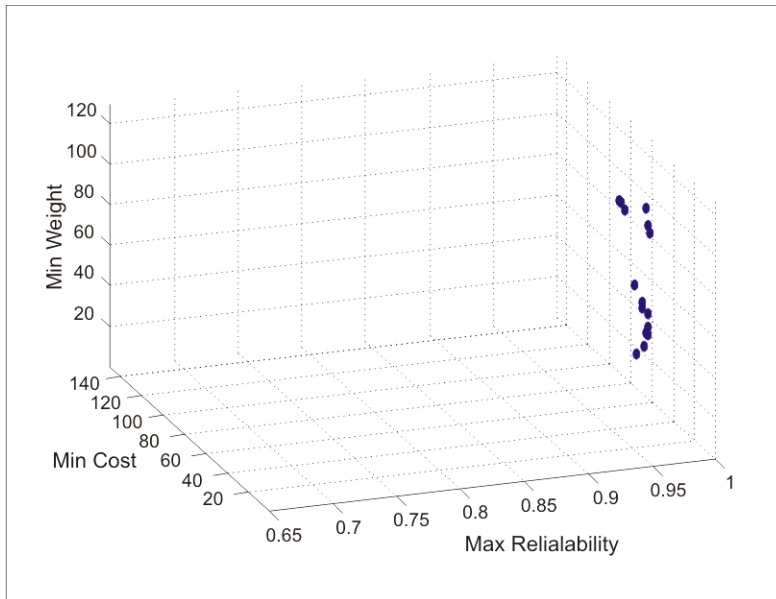


Figura 3. Escenario A Pareto reducido AGPE

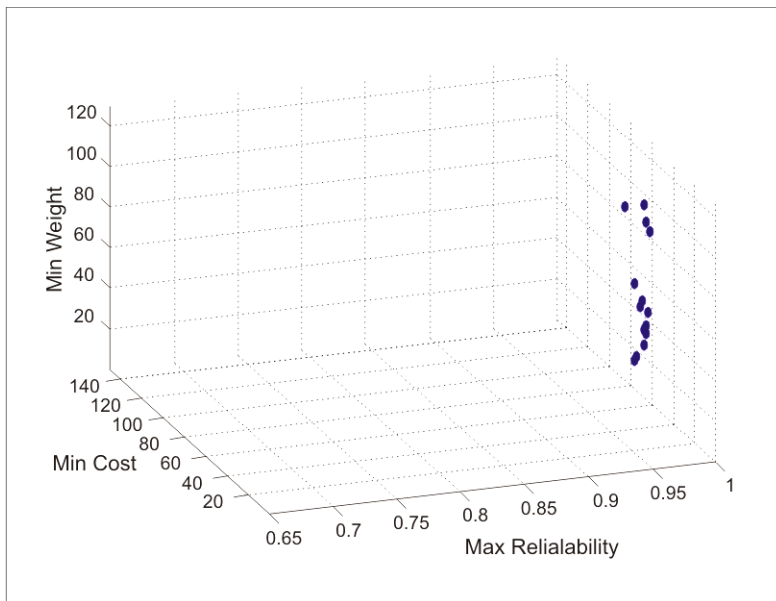


Figura 4. Escenario A Pareto reducido NNR

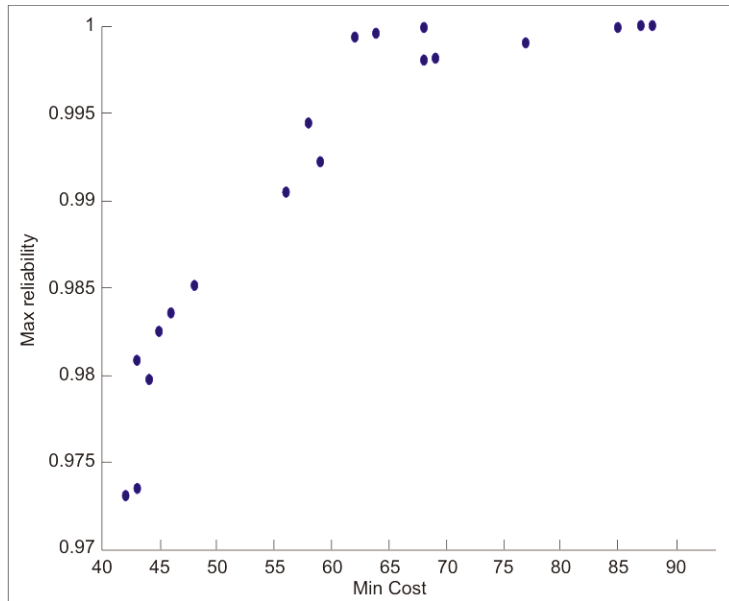


Figura 5. Escenario A Rel. vs. Cost AGPE

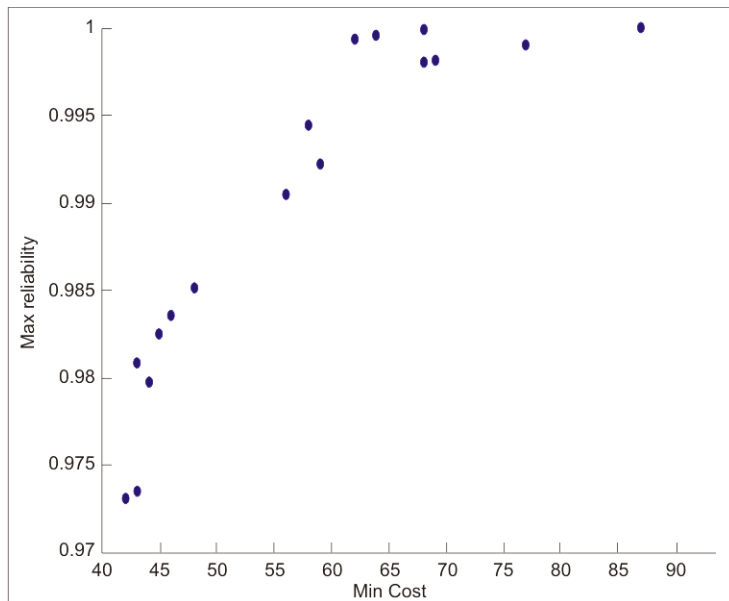


Figura 6. Escenario A Rel. vs. Costo NNR

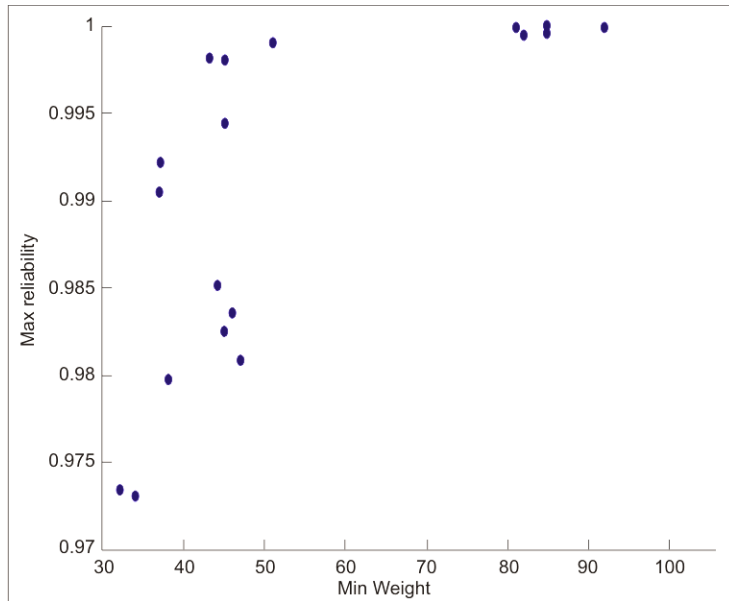


Figura 7. Escenario A Rel. vs. Peso AGPE

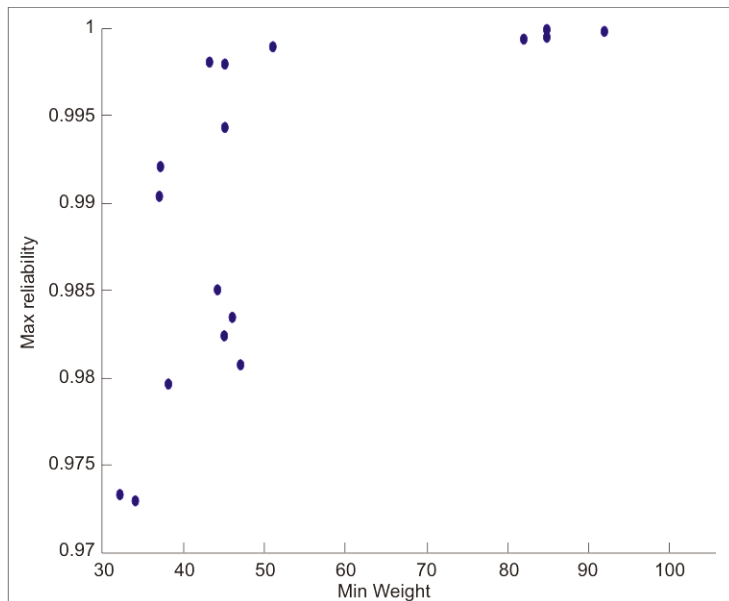


Figura 8. Escenario A Rel. vs. Peso NNR

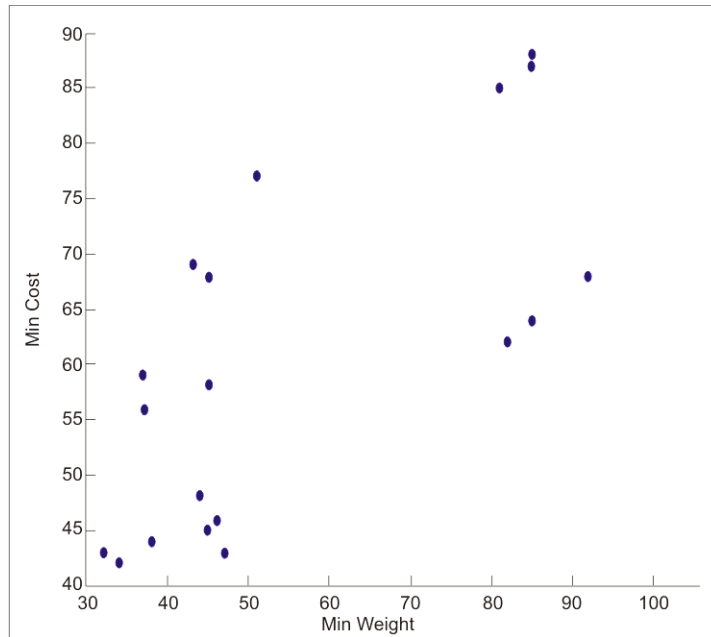


Figura 9. Escenario A Costo vs. Peso AGPE

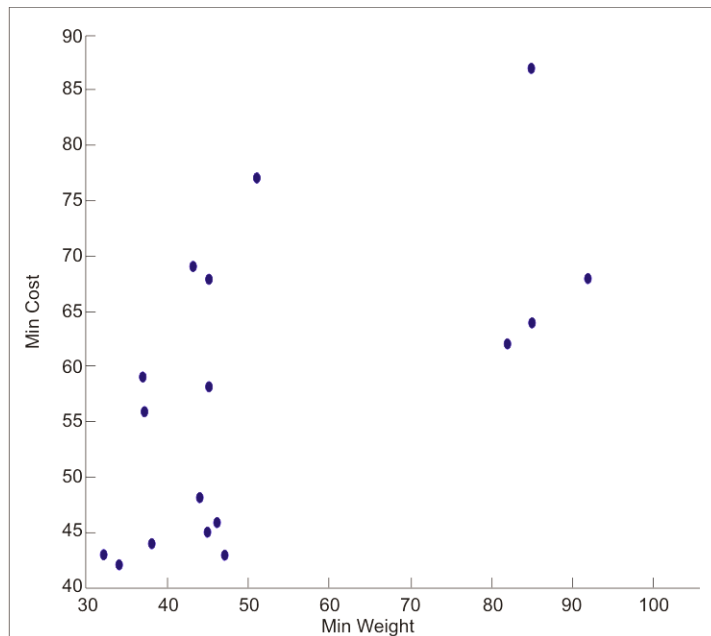


Figura 10. Escenario A Costo vs. Peso NNR

Tabla 3. Escenario A Pareto reducido

AGPE Non-numerical Ranking		AGPE Non-numerical Ranking	
Solution#	Obj. Value	Solution#	Obj. Value
31	0.150633643	31	0.150422476
32	0.150793863	32	0.142520511
34	0.124824989	34	0.136730409
35	0.129582211	35	0.138649513
37	0.117686245	37	0.115190302
40	0.110331525	40	0.110979944
42	0.114825443	42	0.114499019
45	0.111673179	45	0.114120772
47	0.092789061	47	0.082322646
49	0.075071472	49	0.076001521
51	0.050062071	51	0.054692089
52	0.053995122	52	0.075930576
56	0.01440701	56	0.019762642
60	0.064931986	60	0.060513903
62	0.006867112	62	0.011765447
65	0.004221352	65	0.003275881
67	0.005608908	--	--
68	0.004411733	68	0.003728371
69	0.00197964	--	--

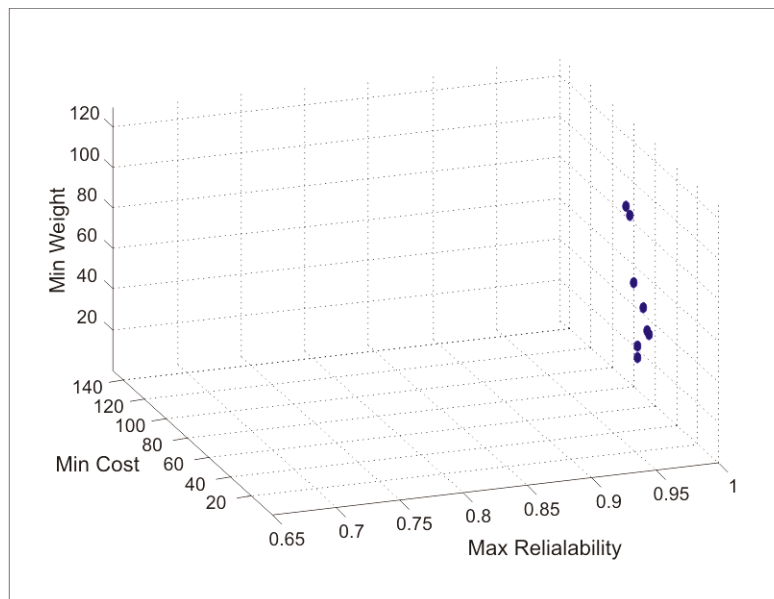


Fig. 11. Escenario B Pareto reducido AGPE

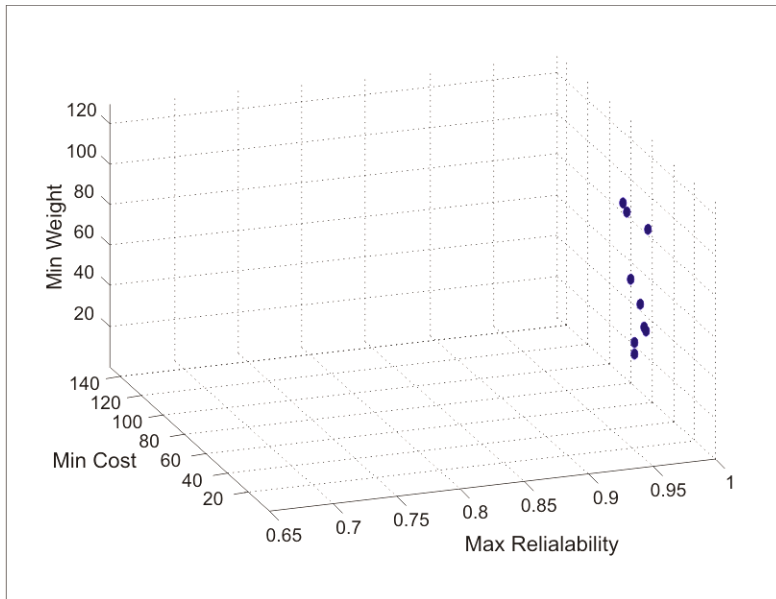


Fig. 12. Escenario B Pareto reducido NNR

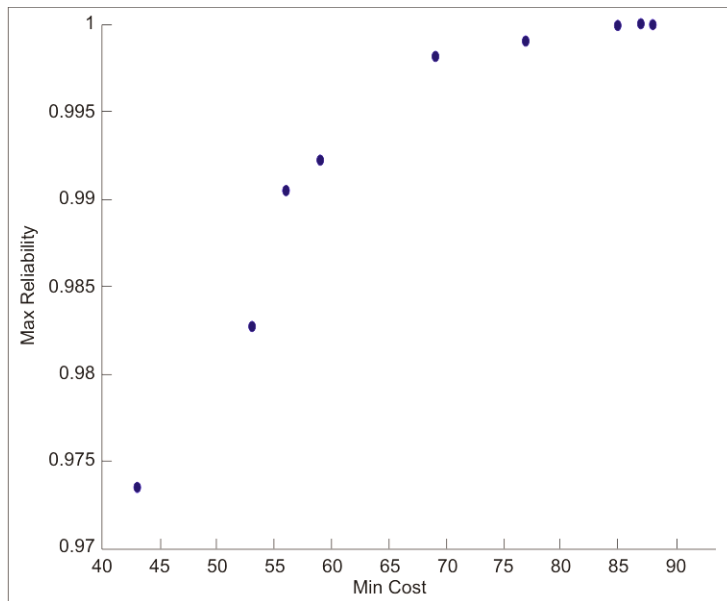


Fig. 13. Escenario B Rel. vs. Costo AGPE

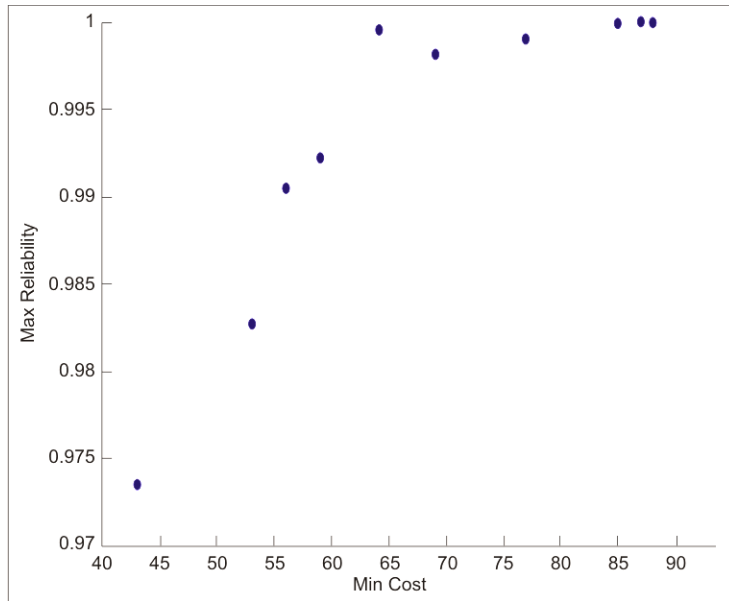


Fig.14. Escenario B Rel. vs. Costo NNR

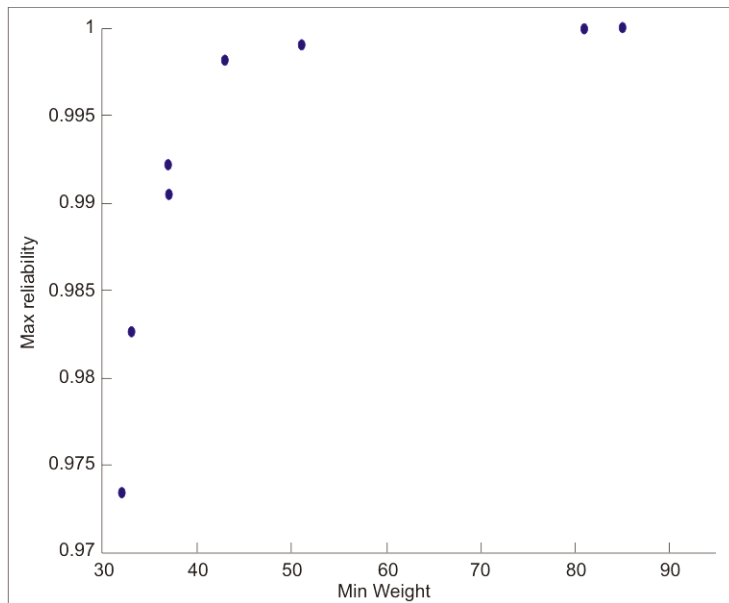


Fig.15. Escenario B Rel. vs. Peso AGPE

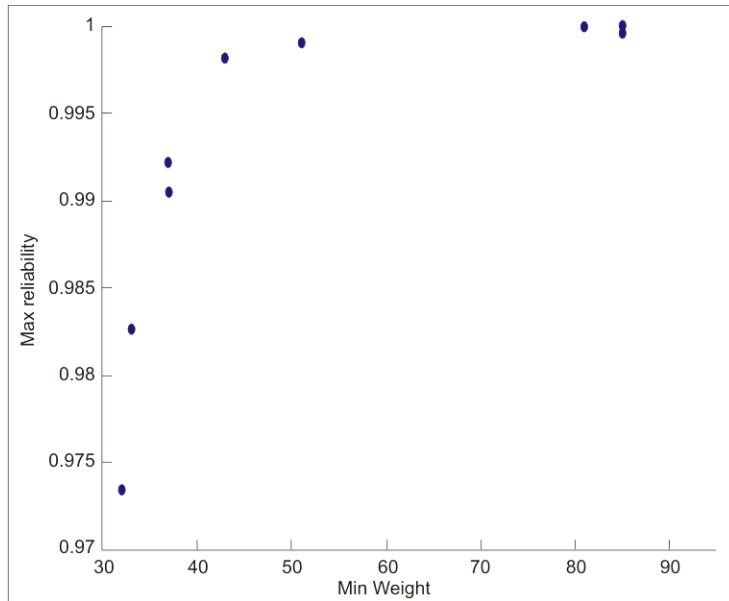


Fig. 16. Escenario B Rel. vs. Peso NNR

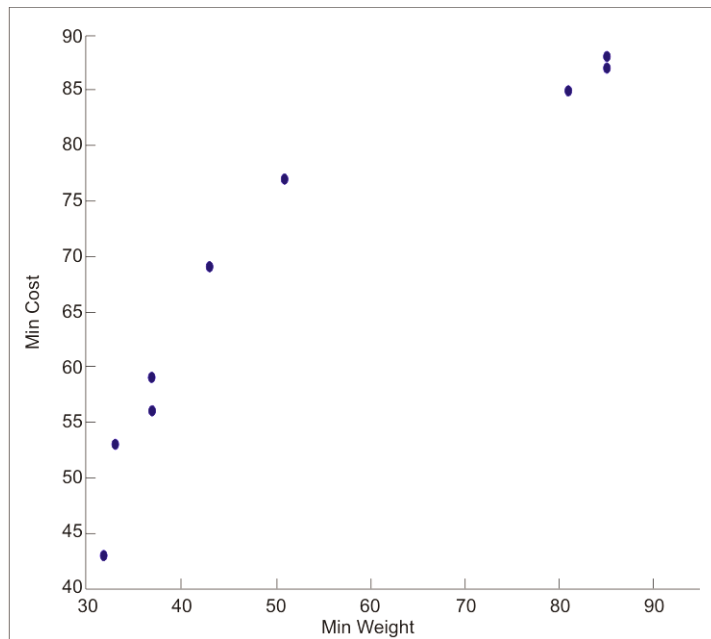


Fig. 17. Escenario B Costo vs. Peso AGPE

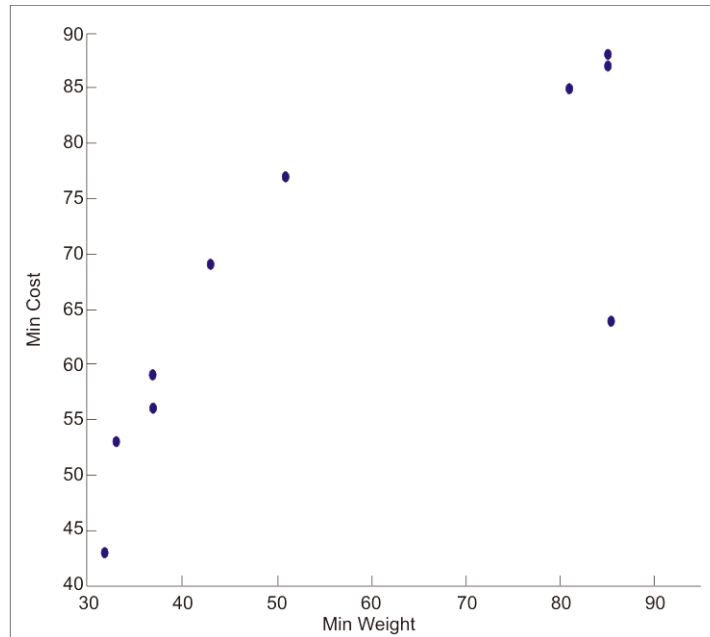


Fig.18. Escenario B Cost vs. Peso NNR

Tabla 4. Escenario B Pareto reducido

IPRG		Non-numerical Ranking	
Solution#	Obj.Value	Solution#	Obj.Value
32	0.144003907	32	0.132440064
38	0.094752871	38	0.108094729
45	0.120303682	45	0.109947915
47	0.089766816	47	0.093817251
52	0.050194032	52	0.042033431
56	0.014624771	56	0.019750443
--	--	62	0.009062697
67	0.006823771	67	0.007685146
68	0.004489561	68	0.003765876
69	0.002696655	69	0.003476225

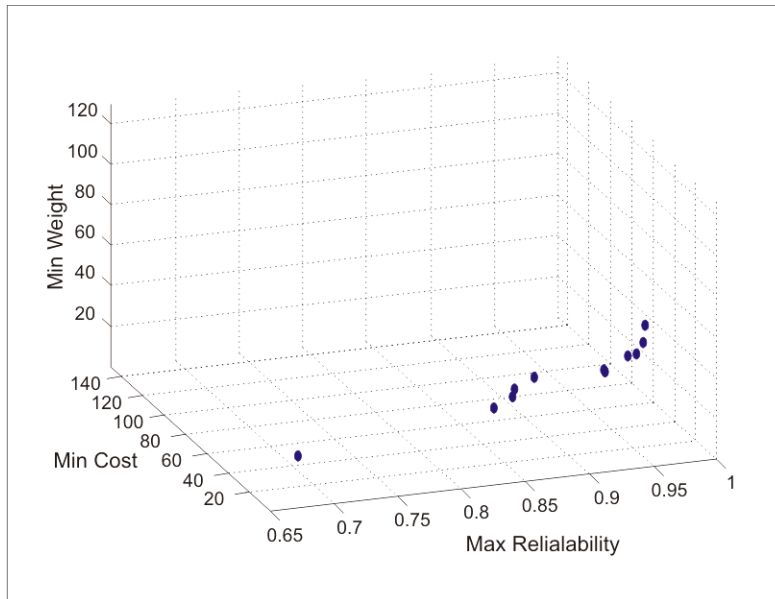


Fig. 19. Escenario C Pareto reducido AGPE

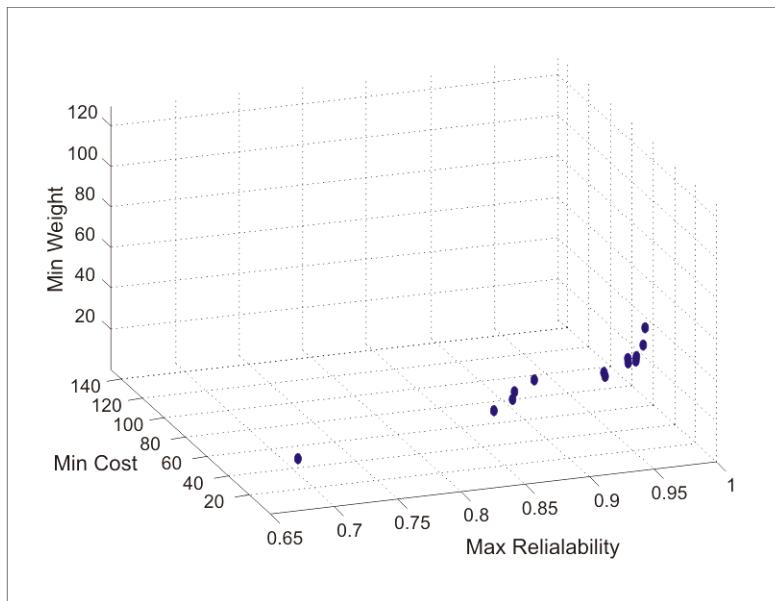


Fig. 20. Escenario C Pareto reducido NNR

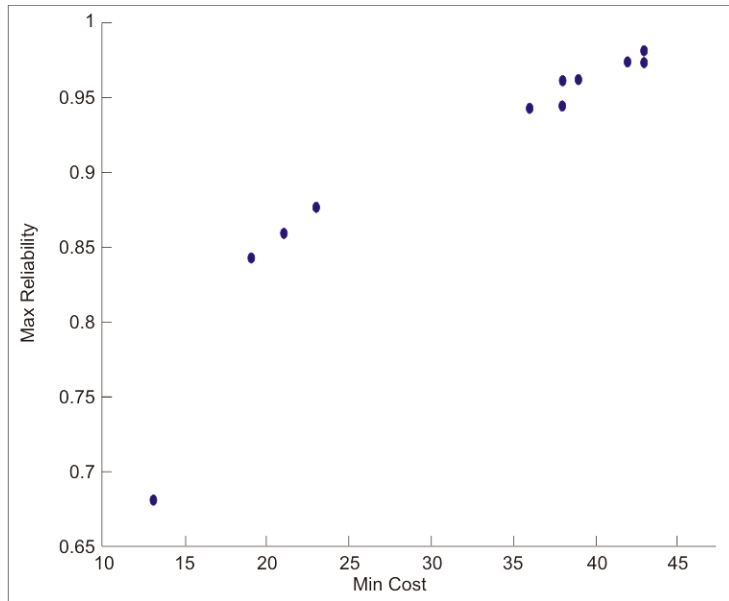


Fig. 21. Escenario C Rel. vs. Costo AGPE

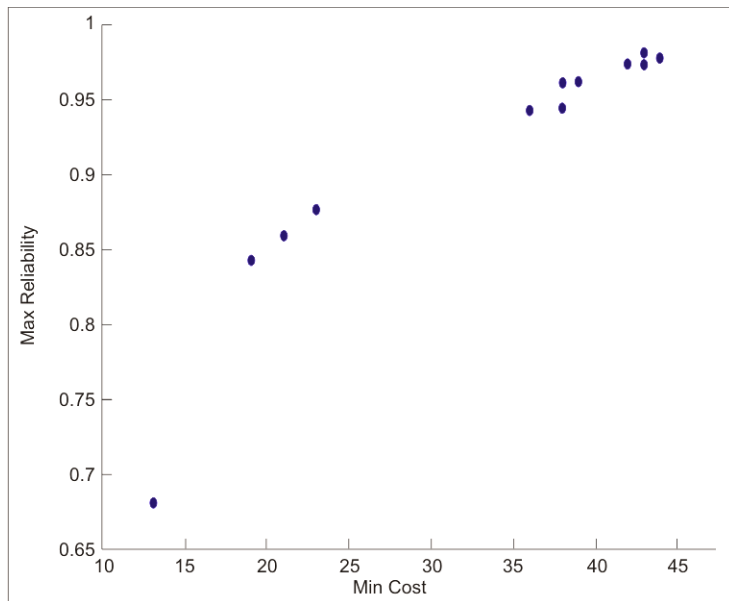


Fig.22. Escenario C Rel. vs. Costo NNR

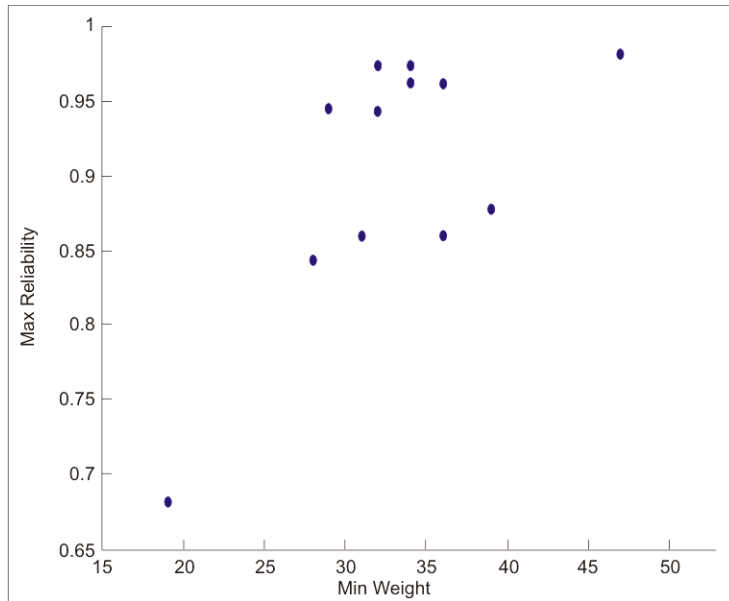


Fig.23. Escenario C Rel. vs. Peso AGPE

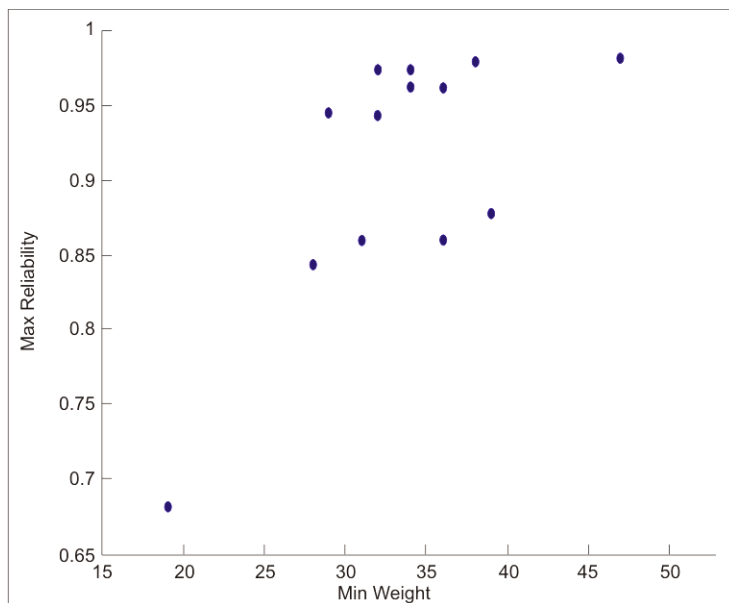


Fig. 24. Escenario C Rel. vs. Peso NNR

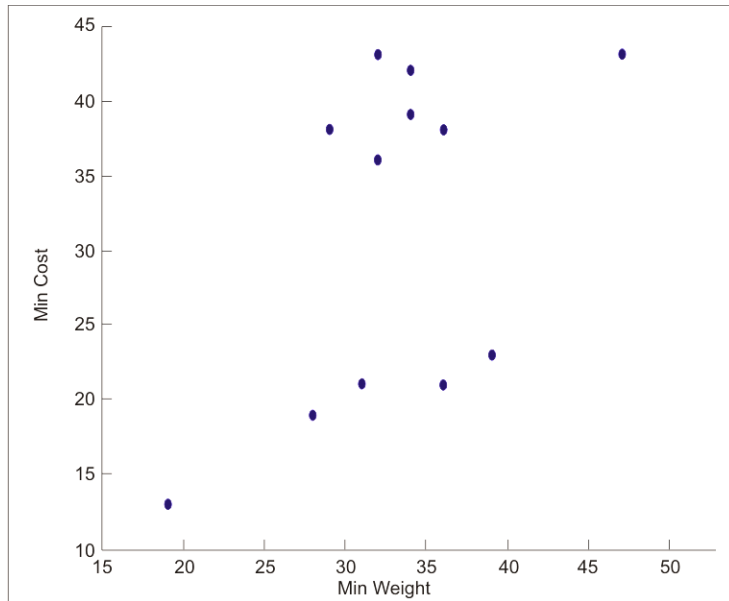


Fig. 25. Escenario C Costo vs. Weight AGPE

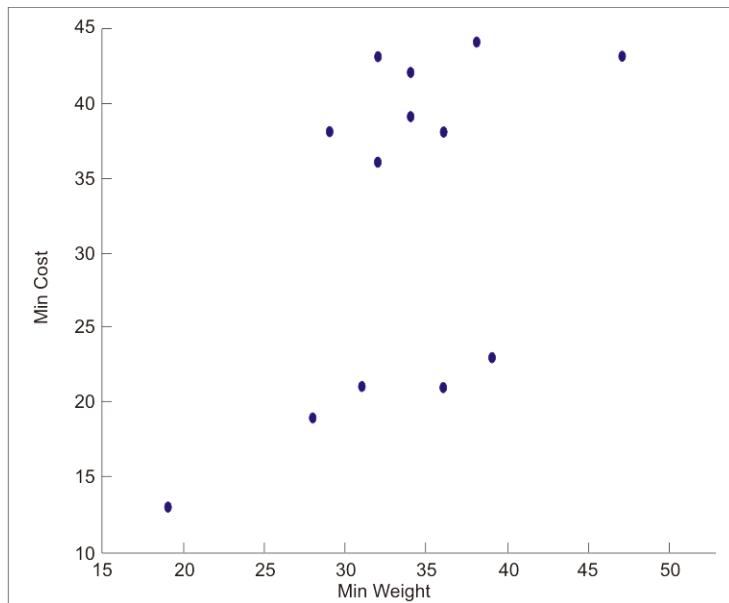


Fig.26. Escenario C Costo vs. Peso NNR

Tabla 5. Escenario C Pareto reducido

IPRG		Non-numerical Ranking	
Solution#	Obj. Value	Solution#	Obj. Value
1	0.01	1	0.066003581
5	0.147467986	5	0.113596013
6	0.169736773	6	0.168516462
7	0.147304029	7	0.14768208
10	0.164708953	10	0.167080579
23	0.167338698	23	0.167376601
24	0.163806109	24	0.164429255
26	0.1677967	26	0.164547458
27	0.167798972	27	0.167547007
31	0.16296138	31	0.156312353
32	0.159535505	32	0.157945676
--	--	34	0.153792145
35	0.154440714	35	0.156982115

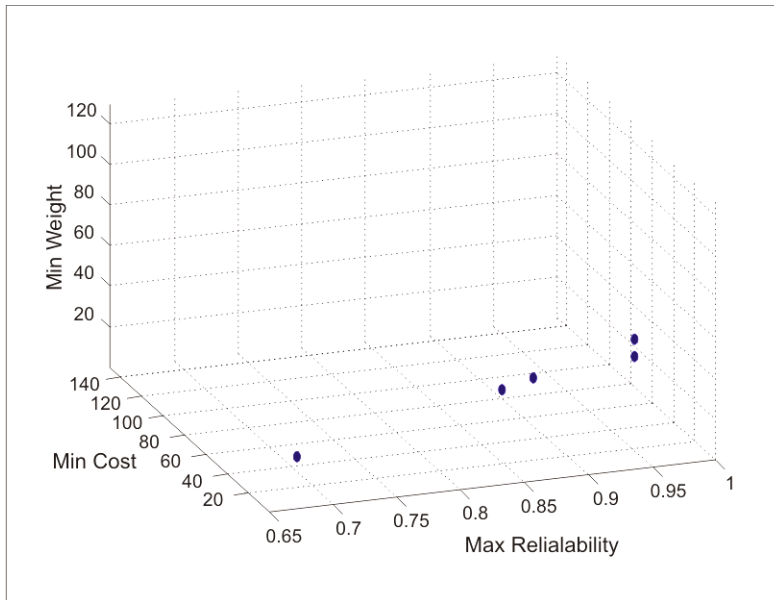


Fig. 27. Escenario D Pareto reducido AGPE

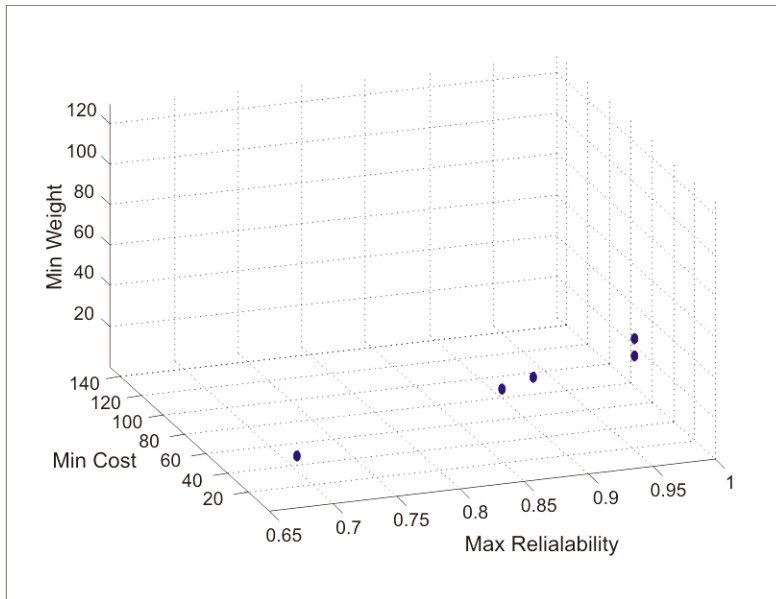


Fig. 28. Escenario D Pareto reducido NNR

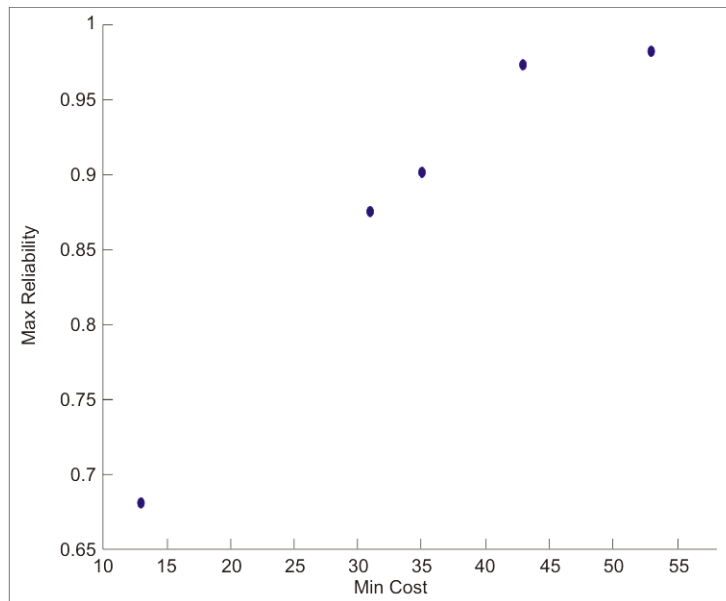


Fig. 29. Escenario D Rel. vs. Costo AGPE

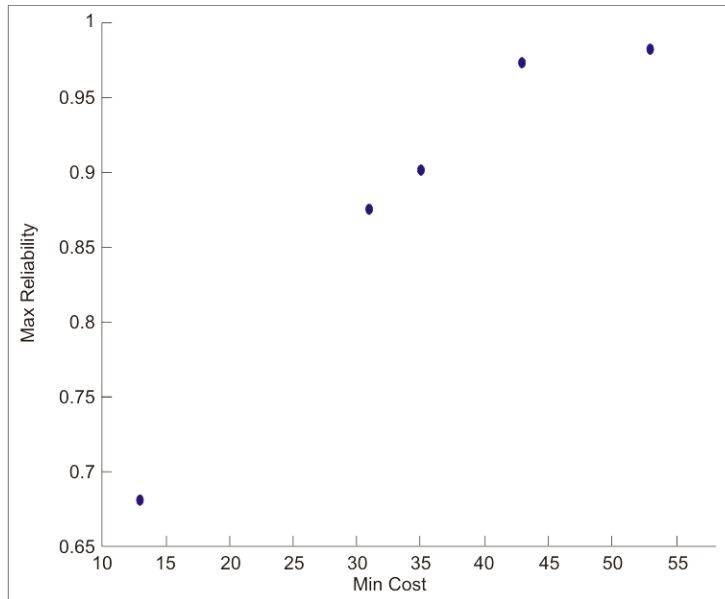


Fig.30. Escenario D Rel. vs. Costo NNR

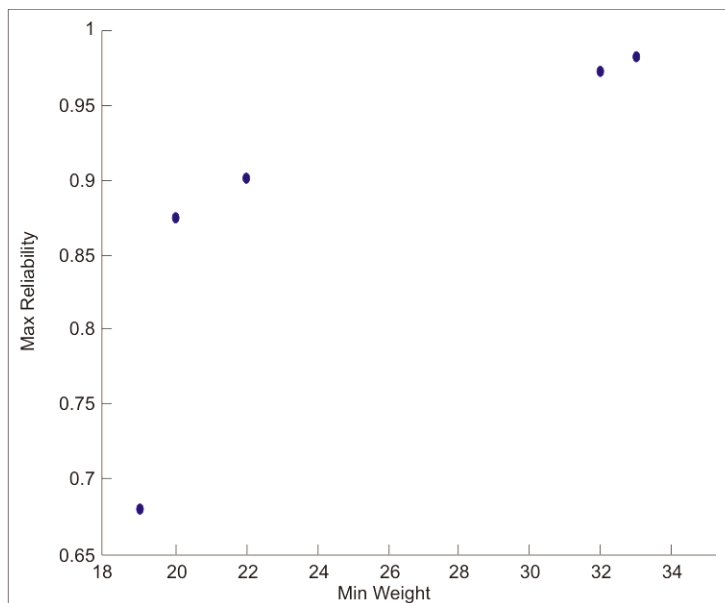


Fig.31. Escenario D Rel. vs. Peso AGPE

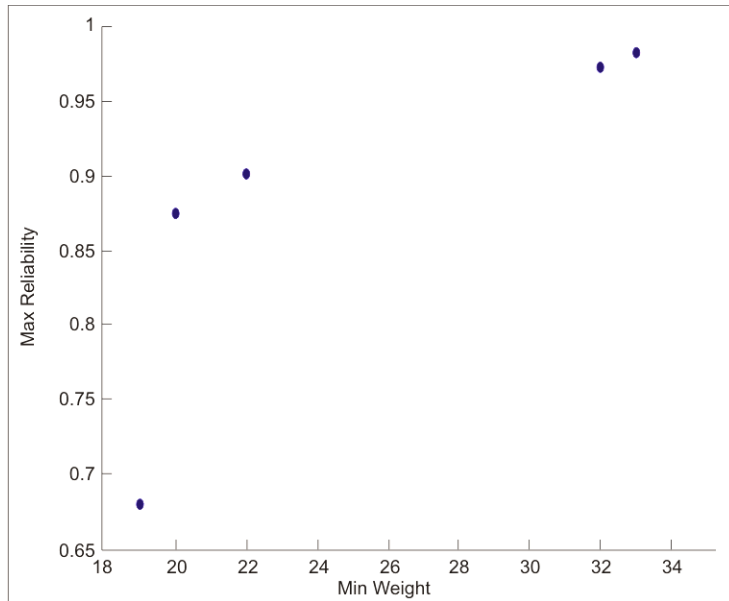


Fig. 32. Escenario D Rel. vs. Peso NNR

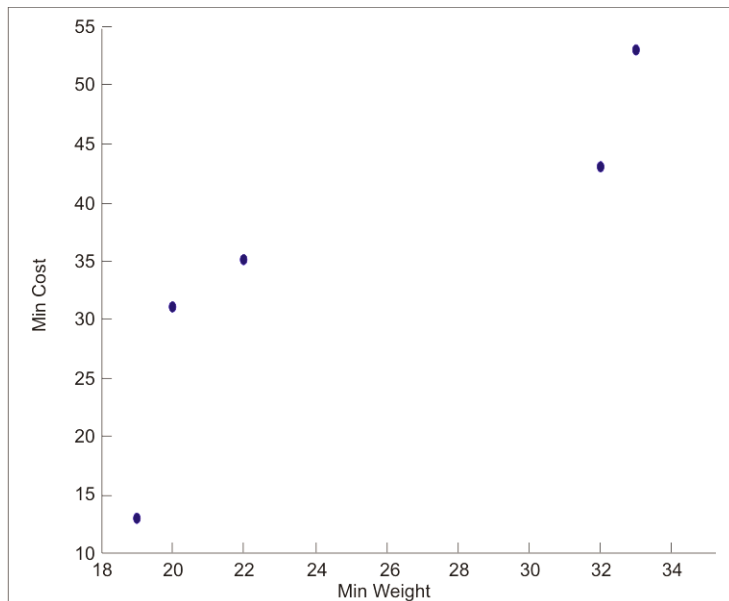


Fig. 33. Escenario D Costo vs. Peso AGPE

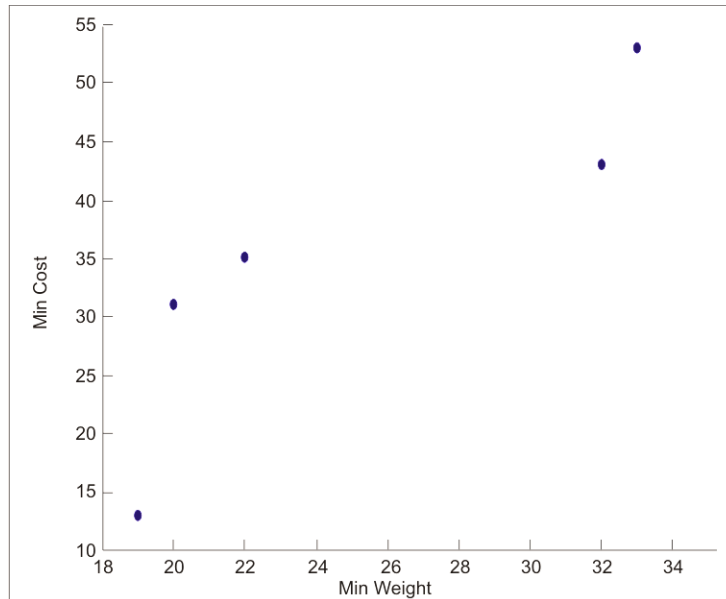


Fig.34. Escenario D Costo vs. Peso NNR

Tabla 6. Escenario D Pareto reducido

IPRG		Non-numerical Ranking	
Solution#	Obj.Value	Solution#	Obj.Value
1	0.003	1	0.017923958
9	0.044111947	9	0.067794536
16	0.112869372	16	0.1081406
32	0.135220832	32	0.122155211
38	0.116847449	38	0.104225983

Frecuencia de las soluciones

Además de obtener una solución de conjunto podado, el número se ha seleccionado una solución en el método se determinó. El método de la poda se puso a prueba durante

5 carreras adicionales en cada escenario. En estas carreras del algoritmo IPRG sirvió como el generador de pesos. Análisis Soluciones repetición de escenarios A, B, C y D se muestran en las Tablas 7, 8, 9 y 10, respectivamente.

Tabla 7. Escenario A Frecuencia de las soluciones en 5 corridas

Run 1		Run 2		Run 3		Run 4		Run 5	
Sol. #	Repetitive	Sol. #	Repetitive	Sol. #	Repetitive	Sol. #	Repetitive	Sol. #	Repetitive
32	2697	32	2764	32	2801	32	2734	32	2659
52	1079	52	1086	52	1051	52	1168	52	1118
60	1060	60	1018	49	1033	34	973	34	1042
34	1023	49	1002	60	1015	60	960	60	1026
49	991	34	998	34	965	49	956	49	964
35	894	35	929	35	882	45	929	35	897
45	885	45	867	45	880	35	895	45	897
62	422	62	391	62	396	62	414	62	412
47	282	47	298	47	300	47	313	47	319
51	214	51	219	31	204	51	204	51	211
31	162	31	168	51	195	31	180	31	163
56	97	65	88	65	91	56	88	56	94
65	88	56	74	56	84	65	82	65	86
68	40	68	32	68	33	68	36	42	33
42	24	42	27	42	23	42	26	37	32
37	23	69	17	37	22	37	20	68	29
69	17	37	16	69	22	69	17	69	11
67	2	67	5	67	2	67	3	67	6
		40	1	40	1	40	2	40	1

Tabla 8. Escenario B Frecuencia de las soluciones en 5 corridas

Run 1		Run 2		Run 3		Run 4		Run 5	
Sol. #	Repetitive	Sol. #	Repetitive	Sol. #	Repetitive	Sol. #	Repetitive	Sol. #	Repetitive
52	2802	52	2832	52	2860	52	2848	52	2906
32	2656	47	2589	47	2686	32	2590	47	2578
47	2595	32	2556	32	2513	47	2544	32	2550
38	1017	38	1040	38	998	38	1046	38	1019
56	429	56	448	56	439	56	447	56	434
45	363	45	408	45	374	45	379	45	385
67	68	69	65	67	71	69	72	67	63
69	67	67	56	69	53	67	69	69	58
68	3	68	6	68	6	68	5	68	7

Tabla 9. Escenario C Frecuencia de las soluciones en 5 corridas

Run 1		Run 2		Run 3		Run 4		Run 5	
Sol. #	Repetitive	Sol. #	Repetitive	Sol. #	Repetitive	Sol. #	Repetitive	Sol. #	Repetitive
5	3339	5	3312	5	3362	5	3353	5	3316
32	1940	32	1937	32	1919	32	1965	32	1937
1	1770	1	1750	1	1785	1	1750	1	1843
26	1509	26	1523	26	1526	26	1502	26	1472
31	447	31	467	31	438	31	446	31	449
10	285	10	261	10	292	10	291	10	295
35	270	35	256	35	256	35	261	35	245
6	191	6	224	35	245	35	256	6	211
27	131	27	140	6	197	6	188	27	114
24	95	24	103	27	135	27	122	24	92
23	22	23	23	24	79	24	103	23	23
7	1	7	3	23	19	23	19	34	2
		34	1	7	3	34	1	7	1

Tabla 10. Escenario D Frecuencia de las soluciones en 5 corridas

Run 1		Run 2		Run 3		Run 4		Run 5	
Sol. #	Repetitive	Sol. #	Repetitive	Sol. #	Repetitive	Sol. #	Repetitive	Sol. #	Repetitive
9	3805	9	3850	9	3803	9	3844	9	3811
32	3183	32	3071	32	3114	32	3048	32	3105
16	1787	16	1825	16	1851	16	1879	16	1820
38	876	38	919	38	934	38	925	38	906
1	349	1	335	1	298	1	304	1	358

Conclusiones

El método presentado se aplicó con éxito a reducir el conjunto de Pareto de un problema de asignación de redundancia multi-objetivo. En comparación con su predecesor, el algoritmo de AGPE proporciona un número similar o igual de soluciones. Sin embargo, para mejorar la precisión del algoritmo de AGPE, aumentando el valor K puede ser necesario en algunos casos. Una ventaja de la AGPE es su facilidad de cálculo debido a que solo

se requiere de generar números enteros. El método proporciona al decisor una colección pequeña de soluciones lo cual simplifica la tarea de selección de soluciones óptimas del frente de Pareto. Investigaciones futuras se llevarán a cabo para probar el algoritmo AGPE con problemas MOP adicionales. Los resultados obtenidos sugieren que el método presentado en este trabajo puede ayudar a los analistas en el análisis post-Pareto de cualquier MOP de cualquier tamaño.

Referencias

- Ando, S., Suzuki, E. 2006. Distributed Multi-objective GA for Generating Comprehensive Pareto Front in Deceptive Optimization. IEEE Congress on Evolutionary Computation pp. 1569-1576.
- Carrillo, V.M., Aguirre, O., Taboada, H. 2011. Applications and performance of the non-numerical ranking preferences method for post-Pareto optimality. Complex Adaptive Systems, Volume 1.
- Coit, D.W., Jin, T., Tekiner, H. 2009. Review and comparison of system reliability optimization algorithms considering reliability estimation uncertainty. IEEE 978-1-4244-4905-7, 49-53.
- Hui, Z., Jixiang, Z., Yuangxiang, L., Weiqin, Y. 2009. MOMS-HDEA: A Multi-Objective Multi-State Hybrid Differential Evolution Algorithm for system reliability optimization design problems. 2009 International Conference on Computer and Communications Security pp. 92-95.
- Konak, A., Coit, D.W., Smith, A. 2006. Multi-objective optimization using genetic algorithms: A tutorial. Reliability Engineering and System Safety 91 pp. 992-1007.
- Kumar, R., Izui, K., Yoshimura, M., Nishiwaki, S. 2009. Multi-objective hierarchical genetic algorithms for multilevel redundancy allocation optimization. Reliability Engineering and System Safety 94 pp.891-904
- Li, Z., Liao, H., Coit, D.W. 2009. A two stage approach for multi-objective decision making with applications to system reliability optimization. Reliability Engineering and System Safety 94, 1585-1592.
- Marler, R.T., Arora, J.S. 2009. The weighted sum method for multi-objective optimization: new insights. Struct. Multidisc Optim. DOI 10.1007/s00158-009-0460-7 Springer.
- Taboada, H., Espiritu, J. and Coit, D.W. 2008. MOMS-GA: A Multi-Objective Multi-State Genetic Algorithm for System Reliability Optimization Design Problems. IEEE TRANSACTIONS ON RELIABILITY, VOL. 57, NO. 1, pp. 182-191.
- Taboada, H., Baheranwala, F., Coit, D.W., Wattanapongsakorn, N. 2006. Practical solutions for multi-objective optimization: An application to system reliability design problems. Reliability Engineering and System Safety 92 pp. 314-322.
- Wang, Z., Chen, T., Tang, K. and Yao, X. 2009. A Multi-objective Approach to Redundancy Allocation Problem in Parallel-series Systems. IEEE Congress on Evolutionary Computation.