



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERIA DE SISTEMAS – INVESTIGACIÓN DE OPERACIONES

**OPTIMIZACIÓN MULTIOBJETIVO DE UN PROCESO AUTOMATIZADO DE PRODUCCIÓN
BASADA EN FUNCIONES DE AJUSTE ESTADÍSTICO.**

TESIS
QUE PARA OPTAR POR EL GRADO DE:
DOCTOR EN INGENIERÍA

PRESENTA:
CARLOS HENRIQUE MARIANO

TUTOR (ES) PRINCIPAL(ES)
ÁNGEL FERNANDO KURI MORALES – IIMAS/UNAM - ITAM SAN ANGEL

COMITÉ TUTOR
JOSÉ JESUS ACOSTA FLORES - FACULTAD DE INGENIERIA - UNAM
SERVIO TULIO GUILLEN BURGUETE - INSTITUTO DE INGENIERIA – UNAM
IDALIA FLORES DE LA MOTA – FACULTAD DE INGENIEIRA – UNAM
MIQUEL ANGEL PIERA i EROLES – U.AUTONOMA DE BARCELONA

MÉXICO, D. F. ABRIL 2013

JURADO ASIGNADO:

Presidente: JOSÉ JESÚS ACOSTA FLORES
Secretario: SERVIO TULIO GUILLEN BURGUETE
1^{er}. Vocal: ÀNGEL FERNANDO KUI MORALES
2^{do}. Vocal: IDALIA FLORES DE LA MOTA
3^{er}. Vocal: MIQUEL ANGEL PIERA i EROLES

Lugar o lugares donde se realizó la tesis: NOMBRE DEL LUGAR

TUTOR DE TESIS:

ÀNGEL FERNANDO KURI MORALES

FIRMA

Agradecimientos

Siempre al terminar un trabajo la primera sensación que se tiene es que él podría haber sido hecho de una manera muy distinta quizás más contundente o profundizada. Más, la única parte de este trabajo que sin sombra de dudas yo no cambiaría sería estos agradecimientos. Primeramente a mi esposa por su apoyo y cuestionamientos que me hicieron caminar, a mi hijo pelo amor y para toda mi familia por su apoyo. A Jorge y Regina por la solidaridad. A Carlos y Gloria por su apoyo. A Paco por las pláticas estimulantes. A las maestras Francis y Ann por su confianza y apoyo. A Esther por su amistad y su importante apoyo con los trámites. A mi colega Irma y a todos mis colegas de curso que compartirán sus incertidumbres y esperanzas. Al Dr. Sérgio Tulio por acompañarme en mis primeros pasos. A Dra. Idalia, Dr. Jesús Acosta y Dr. Aceves por el apoyo y primeras orientaciones. A la Dra. Mayra por sus palabras de incentivo. A mis amigos de la sesión escolar. A mis colegas del Departamento Académico de Electrotécnica de la Universidad Tecnológica Federal de Paraná UTFPR, Brasil. A Gabriel Ortiz por su labor, orientaciones y enseñanzas con los códigos. Y agradezco a una gran cantidad de personas que me apoyaran con palabras, y o gestos cuyos nombres hacen parte de una larga lista y que hacen parte de una historia de vida y crecimiento y que jamás los olvidaré. Y especialmente a Dr. Kuri por su paciencia, orientaciones y creencia de que yo sería capaz de trascender mis debilidades y transformar un borrador en una tesis doctoral. Y en especial también a Estefanía y Carlo Pece por su visión crítica y compañerismo que me hicieron concretizar el presente texto.

Resumen

La presente investigación tiene como objetivo el análisis de un problema de optimización combinatoria para la resolución de un proceso de inclusión de redundancias y mejoría de la confiabilidad individual y global de un proceso de producción, con el objetivo de encontrar la mejor configuración de máquinas, escenario de operación, que maximice la confiabilidad global y minimice los costos de mantenimiento y de operación de una línea de producción automatizada trabajando 24 horas diarias. Tales líneas son, en la mayoría de los casos, son caracterizadas por una grande interrelación de sistemas neumáticos, hidráulicos, microelectrónicos y eléctricos. Por estas características estos sistemas son muy complejos y presentan modos de fallas de diversos tipos con distintos grados de intensidad o impacto a lo largo de toda vida útil o tiempo de operación del sistema. Debido a la integración de sistemas de naturaleza distintas y aliado al comportamiento estocástico la modelación matemática de estos sistemas es extremadamente compleja y en algunos casos casi imposible sin el uso de simplificaciones. Tales simplificaciones pueden ser de diversas naturalezas como por ejemplo: considerar el sistema en el régimen continuo no tomando en consideración el régimen transitorio, simplificaciones matemáticas o suposiciones estadísticas como la existencia de datos distribuidos normalmente y que el funcionamiento de los equipos y sus respectivos modo de fallas sean estadísticamente independientes de entre otras. Delante de estas características se propone una estructura de simulación/optimización por medio de un simulador orientado a eventos discretos, Rockwell ARENA 11, conjugado a un algoritmo elitista multiobjetivo, NSGA II para la resolución del problema de optimización mencionado anteriormente.

El texto se organiza de la siguiente forma. En el capítulo 1 presentase los encaminamientos teóricos de la investigación como: definición y delimitación del tema, definición del problema de investigación y las preguntas que lo traducen y de objetivos y metodología empleada. En el **capítulo 2** se presenta el marco teórico dividido en cuatro partes principales. La primera parte con algunas definiciones preliminares relacionadas la estructura simulación/optimización a través de la revisión de las referencias del trabajo culminando con la clasificación de tales referencias en cinco dominios distintos: de los métodos; de las estrategias y tácticas; de la interfaz; de los informes y de las aplicaciones en los paquetes comerciales. Se destaca también la definición de un entorno de

simulación y por fin se presenta, de forma sintética, las metodologías de simulación/optimización y los modelos básicos de optimización basados en simulación. En la segunda parte iniciase con la definición de la optimización multiobjetivo, multicriterio o vectorial y su importancia en el trato de los conflictos existentes en los problemas de optimización reales donde los objetivos compiten entre sí por los mismos presupuestos, materias primas, espacio entre otros. Destacase también puntos especiales propios de la optimización multiobjetivo, enumerase algunos métodos clásicos y se presenta una clasificación. En la tercera parte hablase de la optimización multiobjetivo a través de los algoritmos evolutivos o heurísticos, sus principios, a importante definición de dominancia y frente o frontera de Pareto y destacase también algunas características del algoritmo NSGAll, presentando al final un tabla que contiene un resumen cronológico del desarrollo de los principales algoritmo genéticos multiobjetivos utilizados. En el **capítulo 3** encontrase el diseño metodológico donde se presentan los aspectos generales del modelo de simulación/optimización como la descripción del enlace ARENA 11 y NSGA II, de los archivos de entrada y salida involucrados no proceso de comunicación entre simulador y genético, de las características del proceso de producción simulado; como también presentase una descripción de las variables de entrada y salida, de la cantidad de variables utilizada en los experimentos. Presentase la formulación de los costos relacionados con el proceso de optimización de la confiabilidad considerando la inclusión de equipos redundantes y sus reflejos en los costos totales de mantenimiento y operación del sistema y al final la declaración del problema tanto en su forma mono-objetivo cuanto en su forma multiobjetivo. En el **capítulo 4** presentase algunos detalles acerca de los experimentos desarrollados para el cumplimiento del objetivo principal del trabajo. En el **capítulo 5** presentase un análisis sobre la convergencia de los algoritmos genéticos; el análisis sobre los resultados de cada uno de los experimentos donde fueran analizados los mejores individuos encontrados, las mejores configuraciones de máquinas encontradas y presentadas diversas gráficas y tablas con las mejores soluciones como también un análisis estadístico .En el **capítulo 6** la conclusión general, presentase y un análisis comentando aspectos generales a cerca de los principales alcances y limitaciones de la investigación.

Tabla de contenidos

1. DEFINICIÓN DEL TEMA.....	8
1.1 DELIMITACIÓN DEL TEMA - OPTIMIZACIÓN DE LA CONFIABILIDAD DE SISTEMAS.....	8
1.2 PLANTEAMIENTO DEL PROBLEMA	11
1.3 OBJETIVOS	15
1.3.1 OBJETIVO GENERAL	15
1.3.2 OBJETIVOS ESPECÍFICOS.....	15
1. DESARROLLAR UN MODELO PETRI DEL PROCESO DE PRODUCCIÓN;	15
1.4 CARACTERIZACIÓN DEL PROCESO:	16
2. ANTECEDENTES: MARCO TEÓRICO, CONCEPTUAL Y DE REFERENCIA	19
2.1 CONSIDERACIONES A CERCA DE LA ESTRUCTURA SIMULACIÓN/OPTIMIZACIÓN.....	19
2.2 CLASIFICACIÓN DE LAS REFERENCIAS DEL TRABAJO:	19
2.3 CATEGORIZACIÓN DE LA ESTRUCTURA DE SIMULACIÓN/OPTIMIZACIÓN	22
2.4 PROCESO DE OPTIMIZACIÓN MULTIOBJETIVO (OM) A TRAVÉS DE ALGORITMOS EVOLUTIVOS	25
2.4.1 DEFINICIONES	25
2.4.2 LOS MÉTODOS DE OPTIMIZACIÓN MULTIOBJETIVO – VISIÓN GENERAL	26
3. LOS PROBLEMAS DE ASIGNACIÓN DE REDUNDANCIAS (PAR)	34
4. DISEÑO METODOLÓGICO	38
4.1 DEFINICIÓN DE LOS MODELOS CONCEPTUAL Y DE SIMULACIÓN DEL PROCESO DE PRODUCCIÓN:	38
4.2 FORMULACIÓN DE LAS VARIABLES, OBJETIVOS Y RESTRICCIONES DEL MODELO COMPLETO (SIMULACIÓN/OPTIMIZACIÓN) ..	41
4.3 EL PROCESO DE INTERACCIÓN ENTRE EL SIMULADOR Y EL OPTIMIZADOR	46
4.4 ELECCIÓN DEL MEJOR ESCENARIO DE OPERACIÓN	50
4.5 DECLARACIÓN DEL MODELO DE PRUEBA NO LINEAL, MONO-OBJETIVO Y NO ESTOCÁSTICO.....	50
5. EXPERIMENTOS	54
6. RESULTADOS: ANÁLISIS Y DISCUSIÓN	55
6.1 EXPERIMENTO 1 MONO OBJETIVO	56
6.2 ESTUDIOS DE CONVERGENCIA.....	56
6.3 ANÁLISIS DE LOS RESULTADOS DE LOS EXPERIMENTOS – MULTIOBJETIVO Y MONO OBJETIVO	66
6.3.1 CONSIDERACIONES ACERCA DEL MODELO MONO OBJETIVO.....	66
6.3.2 – CONSIDERACIONES SOBRE LA ESTRUCTURA SIMOPT	69
6.3.3 ESPECULACIONES ACERCA DE LAS FUNCIONES DE AJUSTE ESTADÍSTICO:.....	74
7. CONCLUSIÓN	84
8. REFERENCIAS PRINCIPALES:	86
ANEXO 1 – MODELO PETRI DIAGRAMA Y TABLAS RESUMEN	97
ANEXO 2 RESUMEN DE LOS ARTICULOS CONTENIDOS EN LAS REFERENCIAS.....	99
ANEXO 3 A FAST AND ELITIST MULTIOBJECTIVE GENETIC ALGORITHM: NSGA II	105

ANEXO 4 GRÁFICAS DE LAS FRONTERAS.....	121
--	-----

ÍNDICES DE GRÁFICAS

Gráfica 1 Distribución de las referencias entre los dominios	21
Gráfica 2 Comparación entre los objetivos 1, 2, 3 y 4 720 horas simuladas – 50 generaciones	57
Gráfica 3 Comparación entre los objetivos 1, 2,3, y 4 para 720 horas simuladas - 100 generaciones	57
Gráfica 4 Comparación entre los objetivos 1,2, 3 y 4 para 150 generaciones y 720 horas simuladas - 150 generaciones	58
Gráfica 5 Comparación entre los objetivos 1, 2,3 y 4 para 1440 horas simuladas	64
Gráfica 6 Comparación entre los objetivos 1, 2,3 y 4 para 2160 horas simuladas	64
Gráfica 7 Comparación entre los experimentos mono y multiobjetivos - 720 horas - 50 generaciones	67
Gráfica 8 Comparación entre los experimentos mono y multiobjetivos - 1440 horas - 50 generaciones	67
Gráfica 9 Comparación entre os experimentos mono y multiobjetivo - 2160 horas - 50 generaciones	68

ÍNDICES DE TABLAS

Tabla 1 - Metodologías de simulación optimización	22
Tabla 2 Desarrollo de los AG multiobjetivo	32
Tabla 3 Desarrollo de los AG multiobjetivo segunda generación.....	33
Tabla 4 Distribuciones de falla y reparación y los tiempos de inspección.....	40
Tabla 5 Resultados de los experimentos (1, 2 y 3) mono objetivo.....	56
Tabla 6 Comparación entre los experimentos (1,2 y 3) mono y multiobjetivo - Objetivo 2 - Confiabilidad global del sistema	68
Tabla 7 Mejores escenarios del experimento 4 720 horas 50 generaciones	69
Tabla 8 Mejores escenarios del experimento 5 - 1440 horas - 50 generaciones.....	70
Tabla 9 Mejores escenarios del experimento 6 - 2160 horas - 50 generaciones.....	70
Tabla 10 - Mejores escenarios experimento 7 720 horas - 100 generaciones.....	70
Tabla 11 Mejores escenarios experimento 8 720 horas 150 generaciones	71
Tabla 12 Mejores escenarios hallados después de la agregación de información	72
Tabla 13 - Tabla de resumen estadístico del experimento 4	75
Tabla 14 Coeficiente de simetría de Pearson de todos los experimentos	75
Tabla 15 Tabla de puntuación del teste de Kruskal-Wallis – proba del factor exógeno.....	79
Tabla 16 - Tabla de resumen estadístico para el Objetivo 1 720 horas - 50, 100 y 150 generaciones	82
Tabla 17 - Tabla de puntuación para el teste de Kruskal-Wallis del factor endógeno	82
Tabla 18 Protocolo de análisis estadístico para experimentos de computación evolutiva .	83
Tabla 19 Descripción de las operaciones del sistema	97
Tabla 20 Descripción de los nodos lugar de la RED de PETRI	97

1. Definición del Tema

La presente investigación analiza un problema de optimización de confiabilidad de sistemas que dado el conflicto de los objetivos, una grande cantidad de relaciones sistémicas, la presencia de aleatoriedades y un elevado número de combinaciones de respuestas exige un tratamiento optimizadorio no clásico. Entre los muchos métodos no clásicos existentes, ver [1-2], se opta por la utilización de una técnica de simulación/optimización que es propia para la analice de sistemas donde su complejidad inherente impide la construcción de funciones matemáticas que lo representen. Todo el proceso objetiva la maximización de la confiabilidad global del sistema, minimización de los costos de operación/ mantenimiento y minimización del número de mantenedores necesarios para el área de mantenimiento del proceso productivo analizado.

1.1 Delimitación del tema - Optimización de la confiabilidad de sistemas

Confiabilidad de un sistema representa la probabilidad que él no falle en el cumplimiento de sus funciones o en la realización de una misión en un tiempo y condiciones de operación determinados.

El análisis de confiabilidad de los sistemas, particularmente los industriales, puede ser considerada a partir de dos enfoques: el clásico, enfoque binario, que considera que el sistema sujeto a fallas puede asumir dos estados distintos, entre la total paralización (cero - 0) y el completo funcionamiento (uno - 1) y el enfoque multiestado que considera que el sistema puede asumir diversos estados entre el completo funcionamiento y la total falla [3-4].

El enfoque multiestado caracteriza mejor los sistemas reales, pues estos están sujetos a una serie de factores que producen la disminución de su vida útil [3]. La ocurrencia de fallas en estos sistemas puede no determinar necesariamente la paralización da actividad, a pesar de producir una tendencia creciente a esta paralización.

Delante de esta realidad se emplean políticas de mantenimiento preventivo y predictivo. Estas políticas visan minimizar los efectos de la degradación natural o da forzada por la presencia de fallas indeseables. Fallas disminuyen la capacidad de los sistemas en desempeñar sus actividades principales y aumentan sus costos de operación y de mantenimiento.

Sea binario o multiestado el enfoque adoptado para el análisis y optimización de la confiabilidad de un sistema, esto no interfiere en la elección de la técnica para la optimización de la confiabilidad del sistema que usualmente es realizada según uno de los métodos en la secuela [5-6]:

- Asignación de redundancias – Donde las variables de decisión son los números de redundancias.
- Asignación de confiabilidades – Donde las variables de decisión son las confiabilidades de las componentes o subsistemas.
- Asignación de redundancias/confiabilidad- Donde las variables de decisión son una combinación del número de redundancias y de la confiabilidad de las componentes o subsistemas.
- Asignación de componentes o subsistemas - Donde la configuración del sistema (arreglo de componentes) representa la variable de decisión.

Un problema de asignación de redundancias consiste en insertar en el sistema una cantidad “ x ” de componentes que debidamente asociadas en paralelo con la componente original, proporcionan la mejoría de la confiabilidad del sistema o del subsistema donde será insertada. En otras palabras el objetivo es asignar cierta cantidad de componentes alternativas los cuales serán asociados en paralelo a las componentes originales, proporcionando, así, mejoría de la confiabilidad global del sistema.

Agregado a este procedimiento se debe considerar concomitantemente que la elección de una determinada componente está condicionada también a factores de desempeño individuales como: los menores costos de instalación y de mantenimiento, la confiabilidad individual, peso y volumen de entre otras características. Así un procedimiento de asignación de redundancias al sistema es un problema de optimización de naturaleza eminentemente combinatoria.

Este aspecto combinatorio de los problemas de asignación de redundancias/confiabilidades implica elevados tiempos de computación (NP-duro) [5-6]. Sin embargo, para el caso de problemas reales eso refleja apenas una de las facetas del proceso de modelado/simulación. Problemas reales demandan la observancia de la relación coste/beneficio cuando se busca mejorías en la confiabilidad sistémica.

Convencionalmente, el tratamiento de esta clase de problemas es computacional y simplificador involucrando la maximización de una función de confiabilidad [2]. Esta

función depende de la estructura del sistema, además de los costos, masas y volúmenes asociados cuando da asignación de las componentes alternativas. En estos casos (modelos de baja complejidad) es posible verificar el comportamiento de las fallas del sistema a través de una simulación Monte Carlo. [4]

El enfoque componencial simplifica el análisis de la confiabilidad, pues, en la mayoría de los casos, considera que las componentes asignadas en el sistema son idénticas (forma, función y confiabilidad) y normalmente es empleado para sistemas no reparables de baja complejidad. Luego, desconsidera relaciones más complejas generadas cuando, al revés de componentes, se está asignando subsistemas al sistema original.

Así a la medida que desease explorar problemas reales y consecuentemente una complejidad más grande de las interrelaciones del sistema, se debe abandonar el enfoque componencial. La dificultad en el tratamiento de estos problemas está relacionada no apenas a las interrelaciones, mas también a la cantidad de objetivos a ser optimizados concomitantemente. Así, al revés de la simulación Monte Carlo, se debe optar por métodos de simulación más sofisticados.

La idea de la presente investigación es, por lo tanto, considerar problemas de complejidad superior, más próximos de la realidad. El modelado de estos problemas considera concomitantemente diversos tipos de variables (reales, binarias, estocásticas y no estocásticas), además de más de uno objetivo a ser optimizado. Dada la elevada dimensión/complejidad de los sistemas y de la imposibilidad de su representación analítica optase por un proceso esencialmente simulatorio asociado a otro evolutivo optimizatorio.

Este proceso, en esta investigación, fue aplicado a un caso idealizado, ítem 1.2, delante a las inherentes ventajas del punto de vista académico y de control de las variables coyunturales del sistema. (p.ej. paralizaciones debido a huelgas y falta de insumos). Así, el aspecto práctico esperado es demostrar la viabilidad del proceso al revés de un ejemplo de implementación empírica. El resultado es una metodología pragmática, aplicable a la realidad industrial, particularmente a los ambientes automatizados.

1.2 Planteamiento del problema

Sistemas reales, particularmente automatizados, son caracterizados por una grande interrelación de subsistemas, como, por ejemplo, neumáticos, hidráulicos, microelectrónicos y eléctricos. De esta manera, pueden ser entendidos como conjuntos complejos de baja interferencia humana (automatizados). Debido a esta complejidad, presentan fallas de tipos variados (electrónicas, mecánicas, etc.) con tasas aleatorias. Tales fallas ocasionan la pérdida parcial o mismo total de actividad del sistema.

Considerando, todavía la elevada productividad de estos sistemas, la mitigación de sus fallas tornase una actividad importante, justificando así el uso de operaciones redundantes. Un conjunto de redundancias debidamente insertado puede asegurar la disponibilidad y los niveles de confiabilidad sistémicos.

En esta investigación, considerase un sistema automatizado, representado por un proceso de producción genérico y sin interrupción. El proceso puede contener hasta 45 operaciones, de las cuales 9 constituyen el conjunto básico para el funcionamiento del sistema. Existe también dos almacenes, uno de partes para la producción (izquierda) y otro para almacenamiento de los productos acabados (derecha), figura 1. Existe, todavía, una banda interconectando el sistema y un conjunto de robots.

Cada máquina en el sistema corresponde a una operación distinta, a cual debe ser realizada dentro de una secuencia específica, en el caso, frezado, erosión, ensamble y ajuste. Ya los robots realizan las operaciones de transferencia/almacenamiento de las partes/productos. El sistema completo (máquinas y robots) realiza 9 operaciones, figura 2.

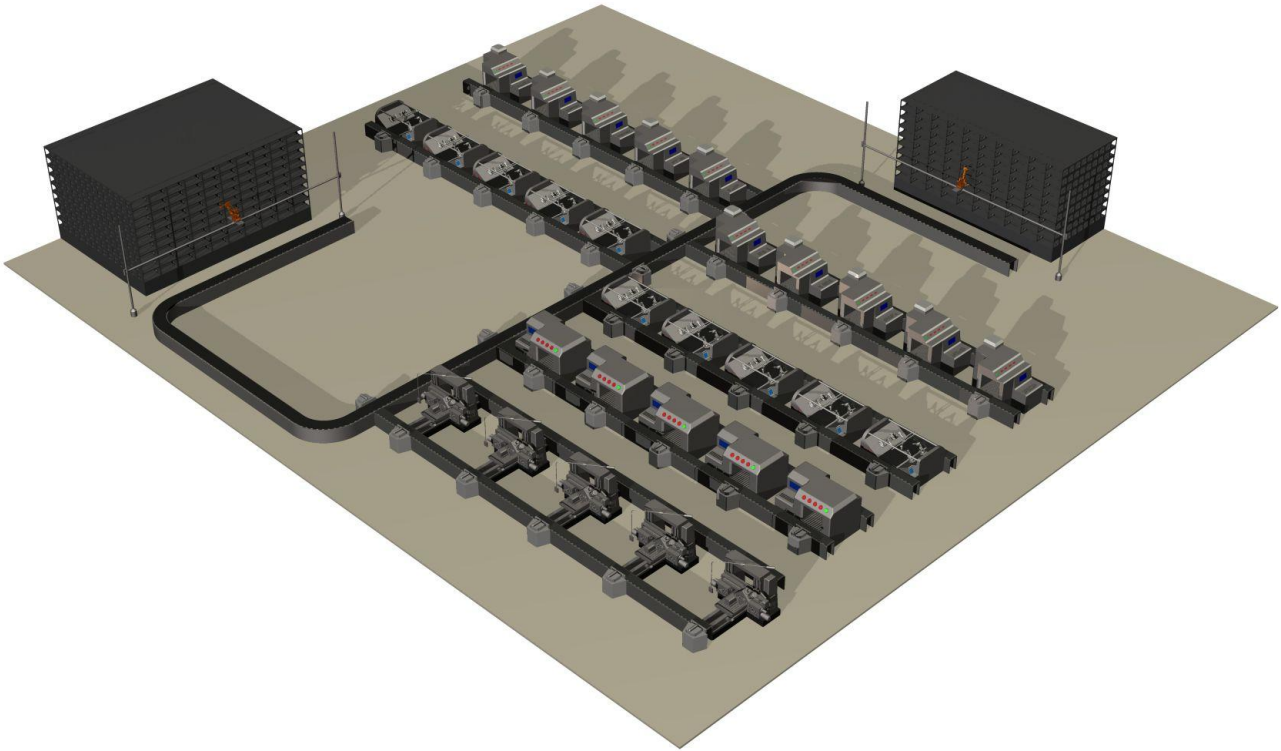


Figura 1 Representación del sistema

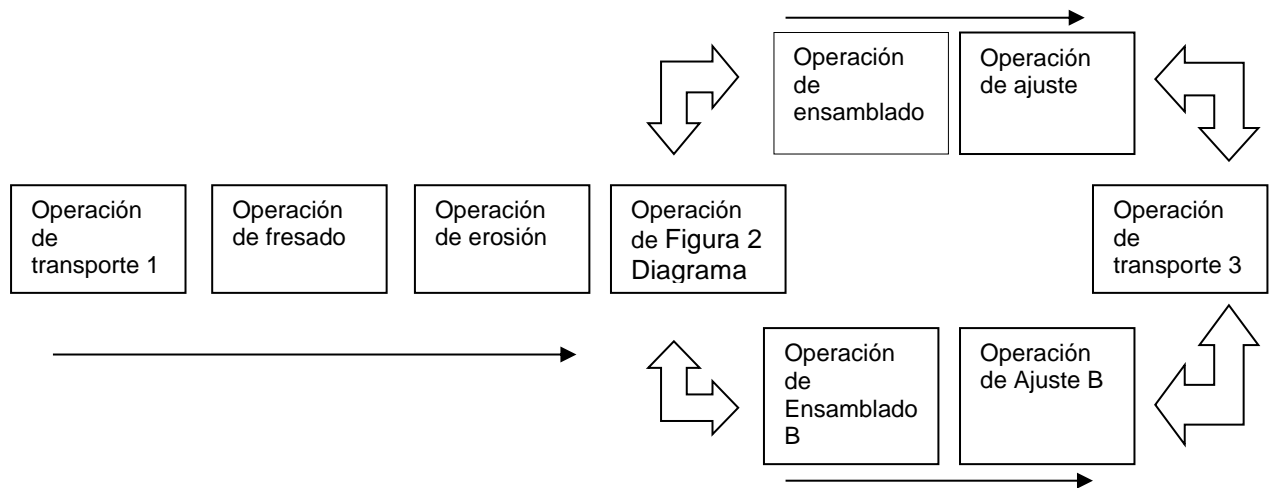


Figura 2 Diagrama de operaciones del sistema

Esquemáticamente, el sistema puede ser representado por uno conjunto de etapas en serie, nueve al todo, correspondiendo a las nueve operaciones básicas del sistema, de acuerdo con figura 2. Cada etapa es ejecutada por hasta 5 máquinas con características

propias de operación. Considerase que apenas una de las 5 sea necesaria para el funcionamiento básico del sistema. Las otras constituyen las redundancias. Es justamente por la utilización/inserción de estas redundancias que es posible incrementar la disponibilidad del sistema. A pesar de la configuración mínima del sistema ser apenas 9 máquinas, el conjunto máximo de 45 posibilita un expresivo número de combinaciones, que necesitaría ser evaluado, figura 3

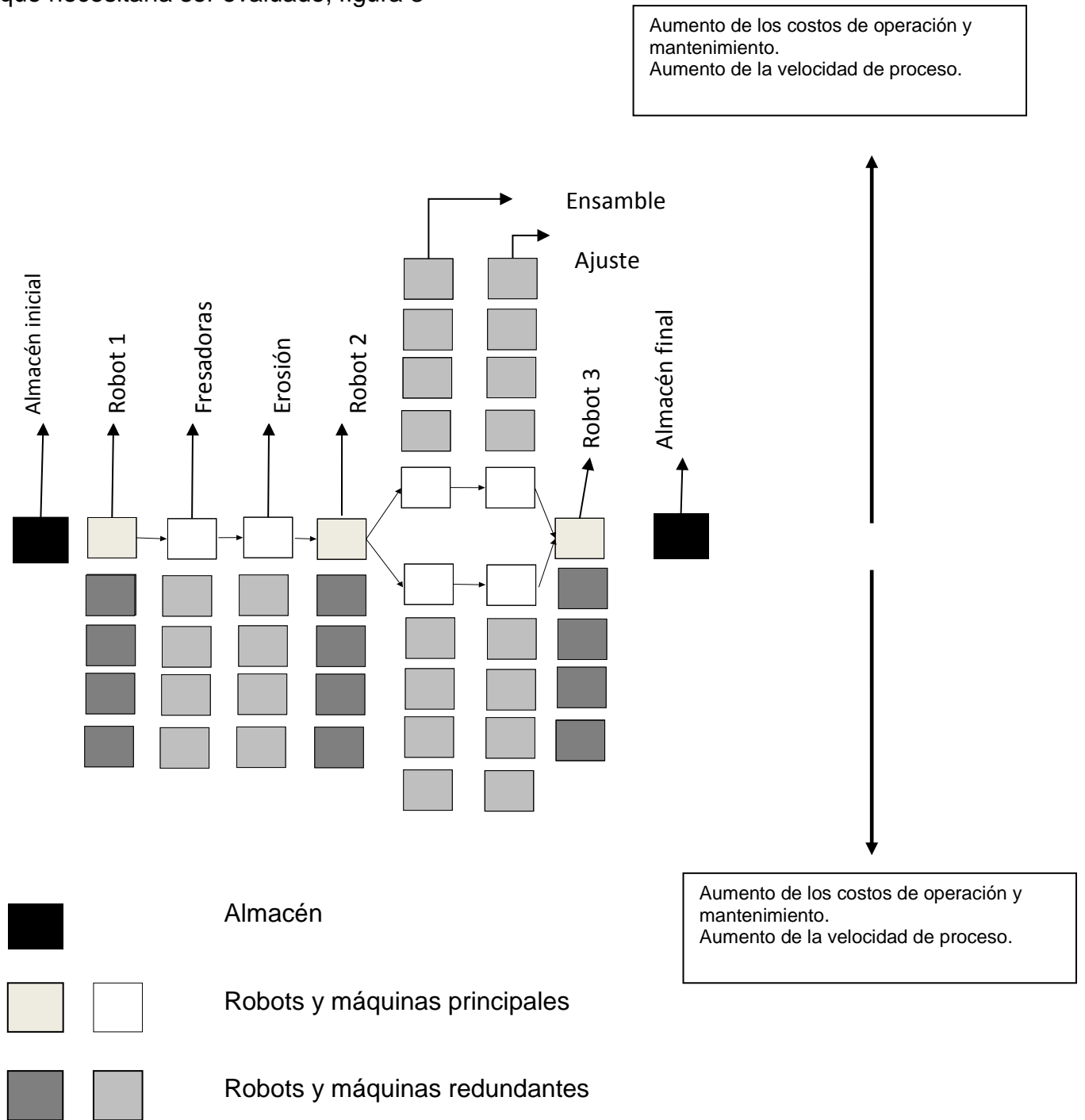


Figura 3 Diagrama esquemático del sistema

Asociado a cada configuración de máquinas/robots, existe un número de mantenedores, los respectivos costos unitarios de operación y fijos de mantenimiento, además de las confiabilidades individuales. A esto conjunto denominase escenario de operación. Es importante observar que escenarios de operación con las mismas características difieren unos de los otros debido la presencia de eventos aleatorios como, por ejemplo, las fallas.

Delante de las consideraciones anteriores el problema aquí abordado puede ser traducido en la cuestión:

¿Cuál será el escenario óptimo de operación para o casa idealizado en cuestión, o sea, aquello que maximiza la confiabilidad total del sistema y concomitantemente minimiza el empleo de mantenedores y los costos de mantenimiento/operación?

La estructura de optimización/simulación propuesta consiste en una técnica de optimización evolutiva multiobjetivo basada en simulación, a través del acoplamiento de un entorno de simulación orientado a eventos discretos, en esto caso ARENA 11, a un algoritmo genético multiobjetivo de segunda generación, el NSGA II. Paralelamente al problema de optimización/simulación propuesto se investiga también las características de convergencia y el proceso de optimización de esto sale dos otras cuestiones de investigación:

- ¿El proceso de optimización funcionará adecuadamente?
- ¿El algoritmo genético convergirá al óptimo global?

1.3 Objetivos

1.3.1 Objetivo general

Investigar experimentalmente el desempeño de una técnica de optimización/simulación multiobjetivo para resolución de un problema combinatorio complejo de asignación de redundancias objetivando la maximización de la confiabilidad global del sistema, minimización de los costos de operación/mantenimiento y minimización del número de mantenedores necesarios para el área de mantenimiento del proceso productivo analizado

1.3.2 Objetivos específicos

1. Desarrollar un modelo PETRI del proceso de producción;
2. Definir el algoritmo genético a ser utilizado;
3. Definir el entorno de simulación a ser utilizado;
4. Definir las variables y los parámetros del algoritmo genético;
5. Definir las variables del proceso simulado;
6. Establecer y programar la interfaz de comunicación entre el algoritmo genético y el simulador;
7. Definir el emulador LINUX para Windows;
8. Establecer un modelo lineal para obtener cotas de comparación y verificación;
9. Establecer las simulaciones;
10. Almacenar los resultados en archivos debidamente identificados;
11. Hacer el análisis estadístico para los experimentos;
12. Hacer la evaluación de los resultados;
13. Escribir un artículo con los principales resultados;
14. Escribir la tesis final;
15. Examen de grado;

1.4 Caracterización del proceso:

El caso idealizado en cuestión presupone la existencia de un sistema automatizado caracterizado por uno escenario de operación (configuración, mantenedores, costos y confiabilidades). Visando la obtención del escenario óptimo y considerando las inherentes dificultades analíticas, aleatoriedades y múltiples objetivos en conflicto que deben ser alcanzados, se propone un proceso de simulación asociado a otro de optimización.

Este proceso hace uso intenso de la computadora, pudiendo ser definido como una técnica iterativa-estocástica. La técnica genera múltiples escenarios de operación, de entre los cuales son elegidos los que mejor satisfagan los objetivos y restricciones del problema. A cada iteración, un conjunto de escenarios/soluciones del problema será elegido de entre una grande cantidad de combinaciones posibles. El proceso de evaluación tiene su inicio en el simulador. Para el sistema automatizado idealizado en cuestión, un sistema discreto complejo, son observadas las relaciones en la secuela:

- Relaciones no lineales
- Relaciones que contienen bucles de realimentación
- Relaciones de intercambio(entradas/salidas) con el medio;
- Relaciones de dependencia del estado anterior;
- Relaciones concatenadas; y
- Relaciones conteniendo variables de naturalezas distintas.

El modelado de uno sistema de esta naturaleza exige un simulador/entorno apropiado. De entre las diversas opciones existentes en el mercado, se eligió el entorno de simulación comercial orientado a eventos discretos ARENA. En este trabajo, adoptase una variante del proceso de simulación/optimización donde el simulador es el corazón del proceso [7-13]. Debido a la elevada dimensión/complejidad del sistema es virtualmente imposible su representación en la forma de una única función de optimización (caso convencional). El modelo de simulación ven, por lo tanto, viabilizar la representación de estas complejas relaciones del sistema.

En esta variante del proceso de simulación/optimización el desempeño del sistema es verificado a través de indicadores/variables que vienen del modelo de simulación. De

esta manera, el simulador produce para el optimizador los subsidios que vendrán nortear la elección de los escenarios (soluciones) más adecuados a los objetivos y restricciones del problema.

De entre los diferentes optimizadores disponibles, se destacan los algoritmos evolutivos. Estos algoritmos manipulan conjuntos de soluciones simultáneamente, posibilitando así la captura de múltiples soluciones en una única iteración. De entre los algoritmos evolutivos, por su vez, se destacan los algoritmos genéticos elitistas (AGE). Estos algoritmos crean una población aleatoria, que evoluciona mediante la aplicación iterativa de operadores de reproducción en la secuela:

- Recombinaciones de los individuos (cruzamiento) y
- Modificaciones aleatorias (mutaciones).

La evolución es guiada por una estrategia de selección que elige los individuos más adaptados a la resolución del problema, o sea, aquellos que mejor satisfagan las restricciones y objetivos del problema. Nótese que la tarea de producir la variabilidad de los escenarios de operación es de responsabilidad del AGE y que esta tarea es realizada de manera automática vía cruzamiento, mutación y selección.

La creación de una generación de individuos/soluciones en el AGE es un proceso estocástico y, por lo tanto, cualquier generación es independiente de las otras. De esta manera, el proceso en cuestión puede ser caracterizado como una secuencia de eventos discretos independientes, o sea, una cadena de Markov, cuya convergencia global es garantizada. De hecho, esta convergencia es garantizada mismo en el caso multiobjetivo, que hace de los AGEs opción robusta y general para los problemas de optimización [14-15].

Cabe resaltar que el tiempo de optimización es aquí utilizado en un sentido más amplio, común a las ciencias de la computación. En este sentido, optimización puede ser definida como un proceso iterativo de búsqueda global donde el óptimo es aproximado vía procesos estocásticos.

El AGE multiobjetivo elegido para esta investigación fue el NSGA II (*Non-dominated Sorting Genetic Algorithm II*) cuyo principio de ordenación es basado en el concepto de dominancia de Pareto [16-17]. En este algoritmo, el proceso de simulación optimización cerraría con la obtención del conjunto de soluciones óptimas de Pareto.

A partir del conjunto óptimo de Pareto, sin embargo, se hace necesario un procedimiento de agregación de información, tanto cuantitativa cuanto cualitativa, para la

tomada de decisión final, o sea, para determinación de la mejor solución. Las informaciones del tipo cualitativo normalmente expresan opiniones de los expertos involucrados en la operación y mantenimiento del sistema. Ya las informaciones cuantitativas son provenientes de las especificaciones técnicas y administrativas.

2. Antecedentes: marco teórico, conceptual y de referencia

2.1 Consideraciones acerca de la estructura simulación/optimización

El proceso de simulación/optimización es una técnica iterativa y estocástica para generación de múltiples escenarios de optimización donde el proceso de optimización ocurre simultáneamente con la simulación del sistema en análisis. De una manera general existen dos enfoques de simulación/optimización el primero presenta la simulación como herramienta de validación o prueba de la efectividad de alguno método de optimización como podemos ver en [18-26] El segundo presenta un proceso de optimización a través de la simulación aplicado a la resolución de un determinado problema como podemos ver en los textos [7-13]. El proceso todo puede ser resumido en dos preguntas: ¿Qué pasa sí?, pertinente a simulación el análisis de múltiples posibilidades. ¿Cómo hago para?, pertinente a la optimización, donde podemos maximizar o minimizar criterios o objetivos importantes para racionalizar la efectividad del sistema.

2.2 Clasificación de las referencias del trabajo:

Considerando el rescate hecho por Bowden & Hall [8] de los requerimientos de Smith para un proceso automático de simulación y optimización, el área de simulación optimización clasificase en seis dominios:

1. La formulación del problema;
2. Los métodos;
3. La clasificación de los métodos;
4. Las estrategias y tácticas;
5. La inteligencia y
6. La interfaz.

Reclasificando los criterios arriba, los 6 dominios, y los aglutinando de otra manera: La clasificación, formulación del problema y inteligencia en un nuevo dominio llamado de Informes (“*Surveys*”) que contiene las revisiones de la materia y presentan en la mayoría de los casos resumen de los tres dominios comentados arriba.

Crease también un nuevo dominio llamado aplicación en softwares comerciales: que trata de la incorporación de los nuevos desarrollos científicos en las áreas de simulación y

optimización a los paquetes comerciales. El abordaje de los dominios, considerando las modificaciones, estará así dividido:

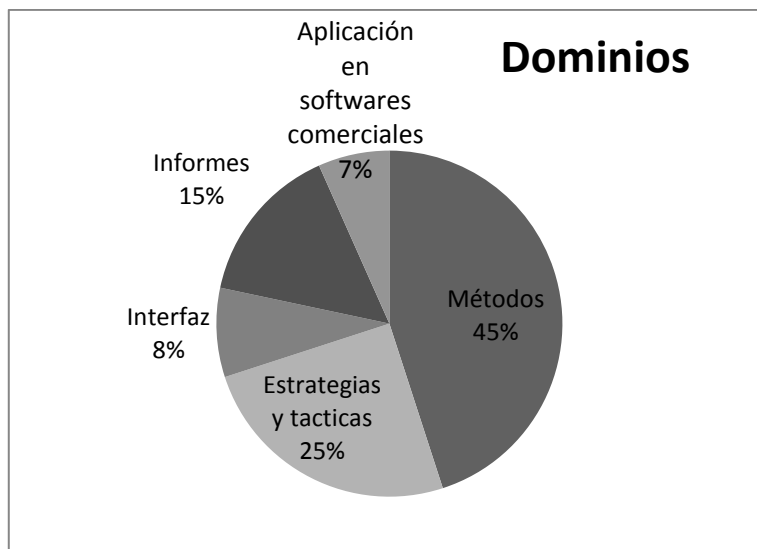
1. De los métodos;
2. De las estrategias y tácticas;
3. De la interfaz;
4. De los informes
5. De las aplicaciones en los paquetes comerciales

Actualmente están disponibles una serie de softwares comerciales que incorporan las últimas tendencias en términos de algoritmos de optimización y procesos de modelado. Son verdaderos entornos de simulación que facilitan el uso de la simulación en la fase de programación del modelo. En la opinión de Guasch, Piera & otros [27] los entornos de simulación actuales ofrecen la misma flexibilidad que los lenguajes de programación, puesto que incorporan bloques que pueden ser programados desde lenguajes de alto nivel como C o Visual Basic. La diferencia en cuanto a presentaciones, capacidades y limitaciones entre simuladores y lenguajes de simulación está prácticamente desapareciendo. Se concluye que la mayoría de las referencias se distribuyen entre los dominios:

- Estrategias y tácticas; y
- Métodos.

Pero existe un límite muy estrecho entre una estrategia y un método o metodología. Pues este último concepto es más amplio que puede contener una o más de una estrategia para la resolución de un determinado problema. Así haciendo un análisis cuantitativo de la tabla del anexo 2 las referencias fueran así distribuidas:

- a. 45% - Están clasificadas dentro del dominio de los Métodos, [28-50] ;
- b. 25% - Están clasificadas dentro del dominio de las Estrategias y tácticas, [51-65];
- c. 8% - Están clasificadas dentro del dominio de la interfaz, [66-70];
- d. 15%- Están clasificadas dentro del dominio de los informes, [7-13]
- e. 7%-Están clasificadas dentro del dominio de las aplicaciones en softwares comerciales, [73-.78]



Gráfica 1 Distribución de las referencias entre los dominios

A partir del dominio de los informes se consideró que la estructura de simulación optimización, en los últimos 10 años ha evolucionado mucho en la dirección de los métodos estocásticos de búsqueda global¹ como también en la dirección de las estrategias híbridas que incorporan el potencial de los algoritmos de búsqueda global a los de búsqueda local.

Hay adaptaciones de métodos de optimización de variables continuas a las variables discretas. Diversos paquetes comerciales de simulación de eventos discretos adoptaran algoritmos de búsqueda global para el proceso de optimización, de acuerdo con [13] y [7], como PROMODEL y *SimRunner*, AUTOMODS, AUTOSCHED y *AutoStat* usan algoritmos genéticos. SIMUL8's *Optimiz* usa redes neuronales. *OptQuest* usa búsqueda tabú, redes neuronales y búsqueda dispersa (scatter search) empleado por los entornos ARENA, CRYSTAL BALL. La razón o razones para eso, según [11], son indudablemente múltiples, pues las propiedades de convergencia tales como convergencia asintótica tienen una relevancia limitada en la práctica, y los meta heurísticos son generalmente rápidos, robustos y generan múltiples soluciones alternativas mientras están enfocados en la búsqueda de una solución óptima. Pero, la aseveración de que la combinación, de tales métodos meta heurísticos y su característica de robustez, con otros métodos que garantizan el desempeño son valiosas y serán parte integral de futuras prácticas de simulación y optimización.

¹ Los meta heurísticos: recocido simulado, algoritmos genéticos, búsqueda tabú.

2.3 Categorización de la estructura de simulación/optimización

Ampliando la categorización hecha por [7] de los métodos empleados por la simulación/optimización² y considerando las contribuciones de [9-13] y [70] la tabla 1 fue hecha y presentada en la secuela:

Tabla 1 - Metodologías de simulación optimización

Tipo de variable de entrada	Categorías		
VARIABLES CONTÍNUAS	Métodos basados en la estimación del gradiente (<i>Gradient base Search</i>) Carson & Maria[18]	<ul style="list-style-type: none"> Estimación De las diferencias finitas (<i>Finite difference Estimation</i>) Estimadores proporcionales de probabilidad o funciones de tanteo (<i>Likelihood ratio estimators or score functions</i>) Análisis de perturbación (<i>Pertubation Analysis</i>) Experimentos en el dominio de la frecuencia o análisis harmónica (<i>Frequency Domain Experiments or harmonics analysis</i>) 	
	Métodos de no gradiente (<i>No gradient based methods</i>) Swishwer[23]	<ul style="list-style-type: none"> Neider-Mead Simples Hooke & Jeeves Camino de muestreo (<i>Sample path method</i>) Simulación reversa (<i>Reverse-simulation</i>) Dilatación de eventos en el tiempo (<i>event Tiempo dilation</i>) 	
	Aproximación estocástica (<i>Stochastic Approximation</i>)		
	Método de las superficies de respuesta (<i>Response surface methodology</i>)		
VARIABLES DISCRETAS	Meta heurísticos (<i>Heuristics methods</i>) o estrategias generales de búsqueda el cambio para meta heurístico es más contemporáneo	<ul style="list-style-type: none"> Recocido simulado (<i>Simulated annealing</i>) Algoritmos genéticos (<i>Genetic algorithms</i>) Estrategia evolutiva (<i>Evolutionary strategies</i>) Búsqueda Tabu (<i>Tabu search</i>) Búsqueda Simplex (<i>Simplex search</i>) 	
	Métodos estadísticos (<i>Statistical methods</i>)	Para un número pequeño de soluciones factibles	<ul style="list-style-type: none"> Selección y tanteo (<i>Ranking and selection R&S</i>) Procedimientos de comparación múltiple (<i>Multiple comparison procedures</i>) Optimización retrospectiva (<i>Retrospective optimization</i>)
		Grande número de soluciones Factibles	<ul style="list-style-type: none"> Optimización ordinal (ordinal optimización) Estrategias generales de búsqueda – meta heurísticos Búsqueda randómica (<i>Random Search</i>)
	Métodos de optimización no para métrica. Estos métodos son empleados cuando, debido a cambios estructurales en el sistema, la función objetivo cambia a cada iteración. Algoritmos genéticos fueran usados con éxito en tales aplicaciones.		
Equipos Asíncronas (<i>A Teams-asynchronous team</i>) Es un proceso que involucra una combinación de varias estrategias de resolución de problemas en una única herramienta. Por ejemplo algoritmos de búsqueda global añadidos con algoritmos de búsqueda local.			

*Según Tompkins & Azadivar [20]:

² Las seis categorías: Métodos de búsqueda basado en el gradiente, Optimización Estocástica, Metodología de la superficie de respuesta, Métodos heurísticos, A-Teams y Métodos Estadísticos

“Los métodos continuos intentan estimar (o usar la estimación) del gradiente de las medidas del desempeño con respecto a las variables de decisión. Ya el termino búsqueda directa implica que ninguna información concerniente a la forma analítica de la función es explotada. En general, los métodos de búsqueda directa no usan la información derivada, ellos requieren solamente el valor de función objetivo para un dado conjunto de valores asignados a las variables de decisión”.

Los problemas de simulación/optimización pueden ser formulados a través de dos caminos alternativos: el primero tratase de un problema de maximización o minimización del valor esperado de la función objetivo de un sistema; el segundo representa maximizar o minimizar la dispersión de la respuesta al revés de su valor esperado [9-10]. En [12] se define una función real que combina todas las variables de salida del simulador dentro de una única variable de salida estocástica el objetivo entonces seria optimizar la función de simulación de respuesta.

La naturaleza estocástica de la salida del proceso de simulación añade complejidad al problema de optimización. Para Carson & María [7] un modelo general comprende n variables de entrada (x_1, x_2, \dots, x_n) y m variables de salida ($f_1(x), f_2(x), \dots, f_m(x)$) o (y_1, y_2, \dots, y_m) , figura 4:

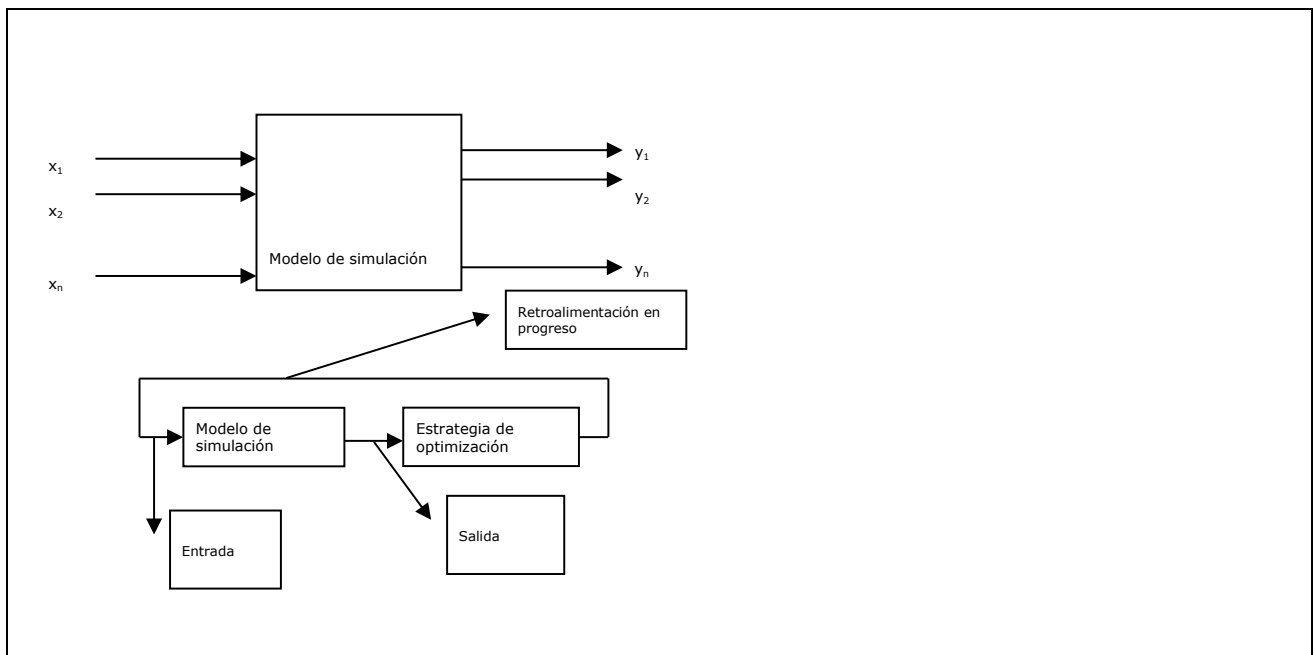


Figura 4 Un modelo de simulación y el diagrama básico de optimización basado en simulación según Carson & María [18]

La figura 5 presenta la compilación de las formulaciones mencionadas anteriormente.

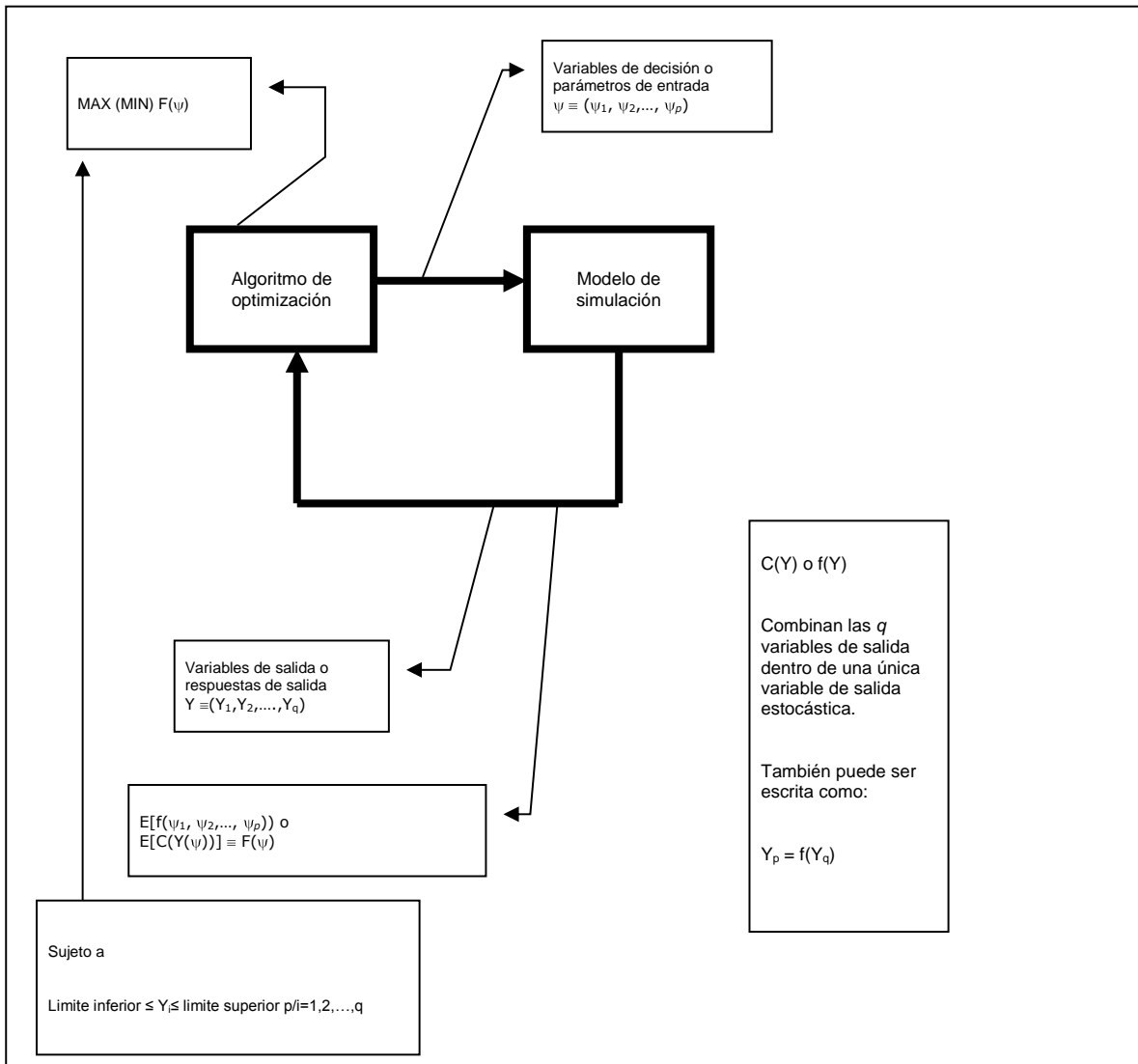


Figura 5 Diagrama del proceso de simulación/optimización considerando las formulaciones presentadas

2.4 Proceso de optimización multiobjetivo (OM) a través de algoritmos evolutivos

2.4.1 Definiciones

Los modelos de optimización multiobjetivo han ganado múltiples aplicaciones en las últimas décadas en diversas áreas de la ingeniería [21], [24], [32], [33], [65] y [78]. Considerando que la mayoría de los problemas en ingeniería son de naturaleza conflictiva donde los múltiples objetivos compiten por los mismos presupuestos, mismos recursos como máquinas e instalaciones, mismos operadores o mantenedores y por la misma estructura del sistema. Así la tarea de maximizar la disponibilidad del sistema, de acuerdo con el caso estudiado en este trabajo, podrá indicar en algunos casos, el aumento de los costos de operación que tan poco son deseados por los analistas del sistema.

Así como un problema de optimización multiobjetivo propone la optimización (minimización o maximización) de un conjunto de funciones, habitualmente en conflicto entre sí, la existencia de múltiples funciones objetivo sugiere la diferencia fundamental entre este tipo de optimización y la optimización mono objetivo: *no existirá una única solución al problema, sino un conjunto de soluciones que invocaran diferentes compromisos entre los valores de las funciones a optimizar*. La formulación de un problema de optimización multiobjetivo y su formulación podrá ser visto en detalles en Nemaschnow [14].

En la secuela presentase un resumen de algunos puntos especiales a cerca de la optimización multiobjetivo:

- El sentido usual de “optimal” no se emplea en el caso de objetivos múltiples debido a que una solución en la que se busque optimizar todos los objetivos simultáneamente, en general, no existe. Al revés, la búsqueda de una solución factible procura el mejor comprometimiento (mejor solución que satisfaga la mayoría de los objetivos) con los objetivos dentro de un conjunto llamado soluciones eficientes (*Pareto optimal, non dominated*);
- La identificación de la mejor solución comprometida requiere que la preferencia del decisor sea expresada siempre considerando:
- Los múltiples objetivos encontrados en los problemas de la vida real pueden frecuentemente ser expresados como funciones matemáticas de varias formas. O

mejor dicho, no solamente nos confrontaremos con objetivos en conflicto sino con objetivos de diversas estructuras.

- Los múltiples objetivos añaden más dificultades a los problemas de optimización combinatoria y son muy difíciles de resolverse exactamente aun cuando sean derivados de problemas simples de optimización con un solo objetivo.

2.4.2 Los métodos de optimización multiobjetivo – visión general

Los métodos clásicos para la optimización multiobjetivo según Gen [15] [78] y Deb [16] [79] se dividen en dos categorías:

1. Métodos de enumeración
2. Métodos basados en preferencia

En el primero se presenta todo el conjunto de soluciones no dominadas al decisor para que el elija la solución que mejor contemple los objetivos. El segundo está basado en una elección de una solución del conjunto de soluciones no dominadas que represente el mejor compromiso y este compromiso es expresado a través de funciones analíticas que van componer los objetivos basados en un determinado grado de preferencia. Deb [16] [79] presenta también otra clasificación o de grano más fino:

1. Métodos de no preferencia
2. Métodos de selección a posteriori
3. Métodos de selección a priori
4. Métodos interactivos.

En realidad la segunda clasificación extiende los métodos basados en preferencia en tres otras sub-categorías. De los métodos basados en preferencia destacamos los métodos:

- a. Método de la suma pesada;
- b. Método de la función de utilidades;
- c. Método de la restricción- ϵ ; (dónde el ϵ representa el límite superior de la función de escenarios medida como restricción del problema);
- d. Métodos de las métricas pesadas;
- e. Método de Benson;
- f. Método de la función valor;
- g. Método de programación por metas;
- h. Método de Pareto;

i. Método lexicográfico

Una descripción más detallada de estos métodos se encuentra en Deb [78] y Gen [79]. Así la complejidad inherente a los problemas de optimización multiobjetivo plantea un difícil reto para resolución mediante algoritmos exactos determinísticos a medida que crece la dimensión del espacio de soluciones. De este modo, las técnicas clásicas enumeradas anteriormente, como también los algoritmos enumerativos o los métodos exactos de búsqueda local, basado en gradientes o que utilizan las técnicas estándar de programación determinísticas – métodos algoritmo glotón (greedy), técnicas de ramificación y acotamiento etc. [14] [80]

Una estrategia que ha ganado muchos adeptos en los últimos años es el uso de algoritmos evolutivos (AEs) que se han popularizado como métodos robustos y efectivos para la resolución de problemas de optimización tanto con uno como para múltiples objetivos.

Los AEs basan su funcionamiento en la simulación del proceso de evolución natural [15]. Consisten en una técnica iterativa que aplica operadores estocásticos sobre un conjunto de individuos (la población) con el propósito de mejorar su grado de adaptación al problema, o desempeño o su *fitness*, una medida, que en la mayoría de las aplicaciones, está relacionada con la función objetivo del problema en cuestión.

Cada individuo de la población representa una solución potencial del problema, codificada de acuerdo a un esquema de representación, generalmente basado en números binarios, reales o ambos en las formulaciones de acuerdo con los algoritmos más contemporáneos.

Así el proceso evolutivo inicia con la creación de una población aleatoria y luego evoluciona mediante la aplicación iterativa de interacciones denominadas operadores de reproducción que incluyen recombinaciones de individuos (cruzamientos) y modificaciones aleatorias (mutaciones). Esta evolución es guiada por una estrategia de selección de los individuos más adaptados a la resolución del problema, de acuerdo a sus valores de desempeño (*fitness*) [14][80].

Ahora los algoritmos evolutivos multiobjetivo, AEMO tienen la potencialidad de tratar problemas con objetivos múltiples, ya que trabajan en paralelo sobre un conjunto de soluciones, hallando en cada ejecución un conjunto de soluciones aproximadas a la frontera o frente de Pareto. Tal característica permite a los AEMOs abordar problemas con espacio de soluciones de grande dimensión.

La frontera de Pareto representa las soluciones que están en las orillas del espacio de soluciones entre los objetivos y su determinación, está basada en la importante definición de dominancia complementando lo dicho anteriormente.

Dice que una solución $x^{(1)}$ domina otra $x^{(2)}$ cuando:

1. La solución $x^{(1)}$ no es peor do que $x^{(2)}$ en todos los objetivos, o $f_j(x^{(1)}) \not\geq f_j(x^{(2)})$ para todo $j=1,2,\dots,M$.
2. La solución $x^{(1)}$ es estrictamente mejor do que $x^{(2)}$ en por lo menos un objetivo o $f_j(x^{(1)}) < f_j(x^{(2)})$, para por lo menos un $j \in 1, 2, \dots, M$.

Sí cualquier una de las condiciones arriba es violada, La solución $x^{(1)}$ no domina la solución $x^{(2)}$. Sí $x^{(1)}$ domina la solución $x^{(2)}$ (o matemáticamente $x^1 \preceq x^2$), acostumbrese también escribirla en cualquier una de las formas siguientes:

- $x^{(2)}$ es dominada por $x^{(1)}$;
- $x^{(1)}$ no es dominada por $x^{(2)}$, o;
- $x^{(1)}$ es no inferior a $x^{(2)}$

Mayores detalles buscar en Deb [6] [16]. La figura 6 siguiente presenta la frontera de Pareto o frente de Pareto que contiene las soluciones no dominadas.

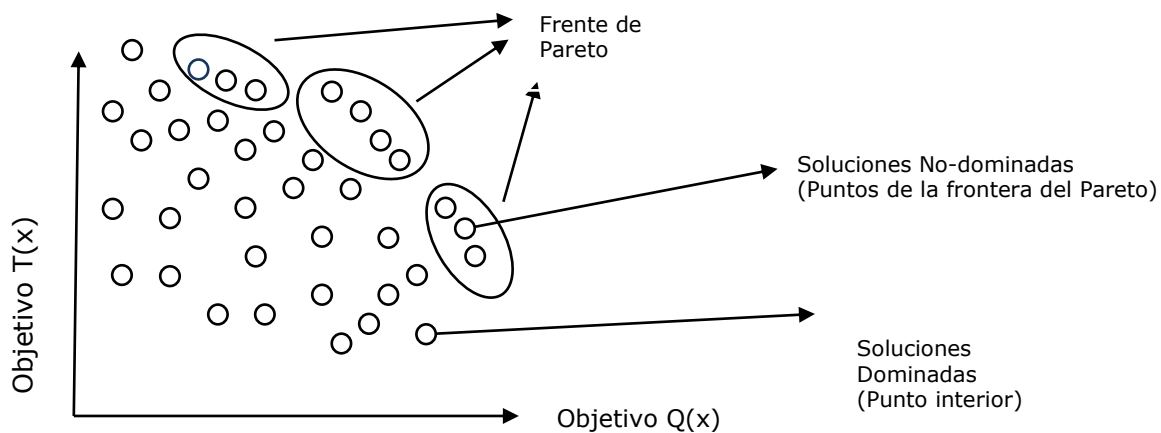


Figura 3 Identificación de la Frontera de Pareto

Para Nemaschnow [80], un AEMO posee dos operadores especiales que no aparecen en la estructura genérica de un AE. Tales operadores son:

A. El operador de diversidad aplica una técnica para evitar la convergencia prematura a un sector de la frontera de Pareto (nichos," *fitness*" " *sharing*", "*crowding*", etc.).

B. El operador de asignación de “*fitness*” orientado a brindar mayor chance de perpetuarse a aquellos individuos con mejores características, considerando los valores de las funciones objetivo y los resultados de la métrica para evaluar la diversidad.

Para este trabajo fue elegido el algoritmo genético multiobjetivo de segunda generación llamado NSGA II.

Non-dominated sorting genetic algorithm II (NSGA-II) es la sigla de uno de los más populares códigos genéticos producidos en los últimos 10 años. Primeramente codificado por Deb, Kalyanmoy y otros en 2002 [16].

Tratase de un algoritmo genético elitista multiobjetivo con complejidad $O(mn^2)$ que proporciona una frontera de soluciones no dominadas, conjunto óptimo de Pareto como visto anteriormente, en una única generación de la población de tamaño n y m funciones objetivo. Tanto el NSGA-II cuanto el SPEA-II [81] (*Strength Pareto Evolutionary Algorithm II*) son exponentes del enfoque basado en el concepto de dominancia de Pareto o abordaje basada en el conjunto óptimo de Pareto que, computacionalmente hablando, ocurre por la formación de una lista de clasificación y/o ordenación o de “*fitness*” de los individuos o soluciones por medio de sucesivas comparaciones considerando que la solución mejor comprometida con todos los objetivos simultáneamente va estar en la cumbre de tal lista.

NSGA-II he estado en el foco de inúmeros trabajos en diversas áreas da ingeniería, como por ejemplo en [82-89]. Otros trabajos exploran desde la más simple comparación de desempeño con otros algoritmos como sus contemporáneos SPEA-2 [108] e μ GA2 [90] como también con algoritmos contiendo mejoramientos que lo rebasan en algún aspecto [91-98]. Además de muchas otras propuestas de modificación o hibridización [99-103]. Como por ejemplo en el trabajo Ishibuchi y Narukawa donde al NSGA-II es incorporado un conjunto de informaciones tipo cualitativas para la disminución del esfuerzo computacional en la búsqueda del mejor conjunto de soluciones [104]. O como en Deb y Tushar donde realizase la fusión del NSGA-II con un algoritmo escalador como mecanismo de búsqueda local [105]. En Pawan y Deb NSGA-II es integrado a una red neuronal [106].

Alteraciones para funcionamiento en paralelo como en Nemaschnow [14] y mejoras de sus operadores de selección, cruce y mutación [107-110] que en muchas veces están enfocados en mejoras marginales sacrificando, en algunos casos, la simplicidad de su código [111].

Pero la popularidad del NSGA-II no está solamente basada en la disponibilidad pública de su código en la internet, más también a su enorme eficiencia pues actúa eficientemente en la manera como establece un orden total en la población y tal orden es uno de los aspectos llaves para lograrse la eficiencia algorítmica principalmente cuando tratase de optimización multiobjetivo [111].

Y por esto que NSGA-II fue, por un largo periodo, considerado el estado del arte en el área de optimización evolutiva multiobjetivo.

Se puede verlo incorporado al Open Beagle (*Beagle – Engine is an Advanced Genetic Learning Environment*) que representa un ambiente de alto nivel o estructura genérica de computación evolutiva destinado a la exploración de la programación genética [112]. Como también el Paradiseo (*Parallel and Distributed Evolutionary Computation framework*) que se trata de una estructura abierta orientada a objetos y dedicada al diseño de metaheurísticos basados en soluciones o basados en poblaciones³[113]. NSGA-II inspiró la creación del SSMO (*Scatter Search algorithm for multiobjective optimization*) incorporado en la plataforma DEME (*DistributEd MEtaheuristics*)[114] dedicada al estudio de problemas de optimización multiobjetivo desarrollada por el personal del NEO (*Networking and Emerging Optimization*) de la Universidad de Málaga España. Figura también en el desarrollo de Sastry como motor de optimización multiobjetivo para el SIMGAT++ (*Single and Multiobjective Genetic Algorithm Toolbox in C++*) [115]. Y también es utilizado en aplicaciones comerciales diversas como por ejemplo la OptinGen library que se trata de una interfaz de programación orientada a objetos que incorpora muchas técnicas de optimización [116].

NSGA II utiliza un mecanismo de preservación de la diversidad muy explícito llamado operador de “*crowding*”, que no requiere parámetros, en vez de usar nichos. Utiliza también un esquema de selección dónde la población de padres se compara con la población de hijos. Además de contar con el uso de elitismo es mucho más eficiente, computacionalmente hablando, que su antecesor NSGA y es un algoritmo altamente competitivo en convergencia a la frontera de Pareto.

Las características principales del algoritmo NSGAII son [14]:

³ Metaheurísticos basados en soluciones corresponden a la familia de algoritmos de búsqueda local que a partir de una solución inicial, generada randómicamente por otro método de optimización, buscan encontrar la mejor solución al problema postulado. Metaheurísticos basados en población corresponden a la familia de algoritmos de búsqueda global que inician su proceso a partir da generación de una población de soluciones candidatas y por medio de operadores especiales evolucionan la población original hasta encontraren la solución mejor adaptada al problema.

1. El ordenamiento no-dominado elitista mediante una técnica de comparación que utiliza un sub-población auxiliar, que le permite disminuir la complejidad de los chequeos de dominancia de $O(MP^3)$ a $O(MP^2)$, donde M es el número de funciones objetivo y P el tamaño de la población utilizada.
2. La preservación de diversidad mediante una técnica de “*crowding*” que no necesita especificar parámetros adicionales.
3. La asignación de valores de “*fitness*” en base a los niveles o rangos de no dominancia se hereda del NSGA original, aunque se considera en el procedimiento de asignación de los valores de distancia “*crowding*” utilizados para evaluar la diversidad de las soluciones.

Para el procedimiento completo del NSGAII ver el artículo en el anexo.

Las tablas 3 y 4, compiladas de [17], en la secuela sitúan cronológicamente el desarrollo de los principales algoritmo genéticos empleados a la optimización multiobjetivo en relación al NSGA II:

Tabla 2 Desarrollo de los AG multiobjetivo

Nombre de los autor(es) y fecha	Nombre de la sigla en Ingles	Comentarios
VEGA – Schaffer/1985 Primero meta heurístico utilizado	Vector Evaluated Genetic Algorithm (VEGA)	Fue una extensión del programa Grefenstette's GENESIS. VEGA selecciona los mejores individuos que satisfagan las condiciones de uno de los objetivos sin mirar para los otros. Esto implica que otros individuos que tengan un mejor desempeño en relación a los otros objetivos podrán no sobrevivir debido a este mecanismo de selección.
MOGA93 por Fonseca y Fleming 1993	Multiple Objective Genetic Algorithm(MOGA93)	En su proceso de "rank" la posición de uno individuo es igual al número de soluciones que dominan o están relacionadas a este mismo individuo.
NSGA por Srinivas y Deb - 1994	Non-dominated Sorting Genetic Algorithm NSGA	El NSGA implementa el proceso de selección de Goldberg en lo cual el "fitness" de un individuo es igual a su área o capa de dominio computado por el "fitness" de la población en la base del dominio.
NPGA por Horn, Nafpliotis y Goldeberg, 1994	Niched Pareto Genetic Algorithm (NPGA)	NPGA combina los principios de la dominancia y del torneo de Pareto donde dos individuos competidores y un conjunto de individuos son comparados para determinar el vencedor del torneo.
MOGA por Murata y Ishibuchi, 1995	Multiple Objective Genetic Algorithm (MOGA)	Este método no está basado en el principio del ranking de Pareto, sino en la suma pesada de las funciones objetivo combinándolas dentro de una función escaladora de "fitness" que usa un generador de valores pesados en cada interacción. Después, los autores acoplaron una búsqueda local con un algoritmo genético, introduciendo el principio del algoritmo "mimético" para problemas multiobjetivos
MGK por Morita, Gandibleux y Katoh, 1998	Method of Morita, Gandibleux and Katon (MGK)	El método distribuye soluciones igualmente voraces o eficientemente soportadas, dentro de una población inicial de tal manera que el algoritmo se inicialice con una buena información genética. El problema de la mochila con dos objetivos fue usado para validar el principio. Este método se transformó en el algoritmo mimético cuando una búsqueda local fue aplicada en cada una de las soluciones potencialmente eficientes.
SPEA por Zitzler y Thiele, 1998	Strength Pareto Evolutionary Algorithm (SPEA)	SPEA toma las mejores características de optimización multiobjetivo de los algoritmos evolutivos y las combina para crear uno único algoritmo. Un problema de la mochila multiobjetivo y con múltiples restricciones fue usado como <i>benchmark</i> para evaluar el método.
PAES por Knowles y Corne, 1999	Pareto Archived Evolution Strategy (PAES)	PAES es una estrategia evolutiva que emplea una búsqueda local para generar una nueva solución candidata y uno archivo de referencia para computar la calidad de la solución
MOGLS por Jaskiewicz, 2001	Multiple Objective Genetic Local Search (MOGLS)	Este método combina un algoritmo exacto con heurísticos de búsqueda local. Uno escalador aleatorio es empleado para seleccionar las soluciones y después las re combina y el producto de la recombinación es mejorado usando un heurístico.
MOGTS por Barichar y Hao, 2002	Multiple Objective Genetic Tabu Search (MOGTS)	Este método es un híbrido en el cual un algoritmo genético es acoplado a una búsqueda Tabú. MOGTS fue evaluado por un problema de la mochila con múltiples restricciones

Algoritmos de segunda generación

Tabla 3 Desarrollo de los AG multiobjetivo segunda generación

Nombre de los autor(es) y fecha	Nombre de la sigla en Ingles	Comentarios
NSGA II	Non-dominated Sorting Genetic Algorithm NSGA. 2000	Nueva versión basada en el NSGA que es más eficiente, computacionalmente hablando, usa elitismo y un operador de comparación saturado que mantiene la diversidad sin la especificación de parámetros adicionales.
SPEA II	Strength Pareto Evolutionary Algorithm (SPEA).2001	Tiene tres diferencias principales del SPEA: <ul style="list-style-type: none"> • Incorpora una estrategia de "fitness" con ajuste fino ("<i>finegrained fitness assignment strategy</i>") que lleva en cuenta, para cada individuo: el número de individuos que lo domina y el número de individuos que son dominados por él. • Usa una técnica de estimación de densidad por vecindad más cercana que, a su vez, guía la búsqueda más eficientemente. • Tiene un método mejorado de corte del archivo que preserva las soluciones de fronteras
μ GA2 Toscano y Coello 2001	Micro Genetic Algorithm 2 - μ GA2	Tratase de una versión revisada del algoritmo μ GA que no requiere ningún parámetro para ajuste fino y propone un esquema dinámico de selección a través de lo cual el algoritmo decide cuál es el mejor operador de cruzamiento a ser usado en cualquier tiempo.

3. Los Problemas de Asignación de Redundancias (PAR)

Los problemas de asignación de redundancias (PAR) están básicamente relacionados con la búsqueda de una determinada combinación de componentes alternativas que, debidamente asociadas a las componentes cuya confiabilidad desease mejorar, proporcionan la mejoría de la confiabilidad global del sistema. Así, son de naturaleza eminentemente combinatoria que pueden ser considerados como un caso especial de la programación entera y, como tal han sido resueltos con métodos tradicionales de corte o de búsqueda o una combinación de ambos como se puede ver en [121],[122] o como problemas de programación no lineal entera-mista en ambos los casos con y sin restricciones. Los PAR son tradicionalmente resueltos en las formulaciones mono o multiobjetivo y más recientemente observase la exploración de los PAR multinivel y multiestado como se puede ver en [123], [124], [125], que agregan un grado aún más grande de complejidad a los PAR tradicionales estudiados y es en esta dirección que los trabajos futuros en el área de los PAR camina (multinivel y/o multiestado).

Para la resolución de un PAR observase el creciente uso de procesos de hibridación de algoritmos combinando métodos heurísticos, redes neuronales, técnicas difusas y algún método de búsqueda local con todos los tipos de meta heurísticos para mejorar la eficiencia computacional o con métodos exactos para reducir el espacio de búsqueda además de la posibilidad de combinarse dos meta heurísticos como los algoritmos genéticos(AG) y el recosido simulado como se puede observar en [126],[127],[128],[129],[130],[131],[132] y [6].

Destacase en esto punto que los algoritmos evolutivos congregan características interesantes que los hacen capaces de manejar los PAR tanto en formulación de uno como de múltiples objetivos y/o multiestado, pues, pueden manejar espacios no continuos, no convexos y/o no lineales tanto cuanto las funciones objetivo que no son explícitamente conocidas. Por lo tanto, los métodos actuales para la resolución de los PAR siguen en la dirección de la aplicación de los algoritmos evolutivos diferentes dos conocidos AG, a pesar de la elevada frecuencia de su aplicación a los PAR, como por ejemplo en [133],[134],[135], y [136].

Observase que en los muchos artículos verificados anteriormente en este ítem que para la prueba de los algoritmos aplicados a la resolución de un PAR multiobjetivo utilizase la formulación general donde se busca maximizar la confiabilidad del sistema,

minimizar los costos asociados a la asignación de componentes como también minimizar la masa (weight) e el volumen del sistema sujeto a restricciones de costo, volumen, masa y de los límites relacionados con la cantidad de componentes que se hacen necesarias como también de los límites de la confiabilidad del sistema. Matemáticamente esta formulación puede ser vista en la secuela:

$$\mathbf{Max} R_s = \prod_{i=1}^m \left(1 - \prod_{j=1}^n r_{ij}^{x_{ij}} \right)$$

$$\mathbf{Min} C_s = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

$$\mathbf{Min} W_s = \sum_{i=1}^m \sum_{j=1}^n w_{ij} x_{ij}$$

$$\mathbf{s. t} \quad \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \leq C_o$$

$$\sum_{i=1}^m \sum_{j=1}^n w_{ij} x_{ij} \leq W_o$$

$$x_{ij} \leq n_{max} \quad i = 1, 2, \dots, m$$

$$x_{ij} \geq n_{min} \quad i = 1, 2, \dots, m$$

$$x_{ij} \in Z^+, \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, n$$

Donde

m – número total de subsistemas o etapa

i – índice do subsistema, $i=1, 2, \dots, m$

j – índice de las componentes en cada subsistema, $j=1, 2, \dots, n$

r_{ij} – confiabilidad de la componente j en el subsistema i

c_{ij} – costo de la componente j en el subsistema i

w_{ij} – masa de la componente j en el subsistema i

R_s – confiabilidad total del sistema paralelo-serie

C_s – costo total del sistema paralelo-serie

W_s – masa total de sistema paralelo-serie

C_o – costo permitido para o sistema

W_o – masa permitida para el sistema

a_i – número de escoja (posibilidad) de componentes para el subsistema i

x_{ij} – cantidad de componentes j usadas en el subsistemas i

n_i – Número total de componentes que pueden estar en paralelo

n_{max} – Número máximo de componentes que pueden estar en paralelo

n_{min} – Número mínimo de componentes que pueden estar en paralelo.

Los PAR pueden ser clasificados cuanto al tipo de redundancias: como de componente pasivo en espera (*cold standby*), como componente activo en espera (*warm standby*) y como componente siempre activo (*hot stanby*) o redundancia en paralelo (*parallel or active redundancy*). Cuanto al tipo de componente a ser asignado: como de componentes idénticas o de componentes no idénticas siendo los últimos más complejos y más próximos de los sistemas reales. A partir del nivel de redundancia de la componente como: redundancia de componente (en nivel de componente individual), redundancia modular (en nivel de subsistema) o redundancia en nivel de sistema (redundancia de sistema). Y, a partir del estado de funcionamiento de la componente como: binario donde la componente puede asumir solamente uno de dos estados posibles, cero para la total falla o uno para el total funcionamiento o multiestado donde entre la total falla y el total funcionamiento el sistema puede asumir múltiples estados. Y por fin también clasícalos a partir de la naturaleza del sistema: como reparable o no reparable.

Para efecto de este trabajo considerase aún el enfoque donde todas las componentes son idénticas que puede ser considerado como componencial básico pues básicamente las componentes son descritas solamente en termos de tres atributos: masa, volumen y confiabilidad individual.

Cuando además de los tres atributos básicos agregase a las componentes otros atributos como: distribuciones de fallas y de reparación, inspecciones, velocidad de proceso y etc. Así se tiene el enfoque componencial complejo. Que agrega un grado de complejidad más grande al problema, de acuerdo con los esfuerzos computacionales requeridos. Tanto el enfoque componencial básico como en el complejo buscase encontrar un valor más adecuado para la confiabilidad individual de la componente que coyuntamente con los otras componentes asignadas proporcionarán el valor total final de

la confiabilidad del sistema, valor esto que depende de la estructura del sistema. En la mayoría de los ejemplos observados los analices fueran hechos debajo una estructura clásica paralela-serie, [6], cuya expresión general para su representación es dada por las ecuaciones:

$$R_i = 1 - \prod_{j=1}^{n_i} (1 - r_{ij}) \quad , \text{ la confiabilidad del sistema es dada por } R_s = \prod_{i=1}^k R_i$$

Donde

R_i – confiabilidad de uno subsistema i

r_{ij} – es la confiabilidad de una componente j , $1 \leq j \leq n_i$ del subsistema i

n_i – componentes en paralelo

k – subsistemas en serie

Si agregamos las dos expresiones tenemos

$$R_i = 1 - \prod_{j=1}^{n_i} (1 - r_{ij}) \quad , \text{ y la confiabilidad del sistema es dada por } R_s = \prod_{i=1}^k R_i$$

Donde

R_i – confiabilidad de un subsistema i

r_{ij} – é a confiabilidad de uno componente j , $1 \leq j \leq n_i$ do subsistema i

n_i – componentes en paralelo

k – subsistemas en serie

Si agregamos las dos expresiones tenemos:

$$R_s = \prod_{i=1}^k \left(1 - \prod_{j=1}^{n_i} (1 - r_{ij}) \right)$$

Más, como el valor de la confiabilidad solo asume su sentido completo cuando es asociado a un determinado intervalo de tiempo, observase en la revisión de los artículos que este aspecto es poco comentado y es en este contexto que la presente investigación esta inserida pues al considerar el enfoque componencial complejo lleva en consideración la influencia del tiempo pues los valores de la confiabilidad ahora son evaluados en función del tiempo de funcionamiento del sistema. Ahora tratase de una componente que en realidad es un subsistema (máquina o equipo) cuyos atributos principales no están solamente relacionados con la masa, el volumen y la confiabilidad individual. Y tal subsistema representa una máquina de proceso que ejecuta una determinada tarea dentro de una línea de producción y, que es caracterizada pelos atributos en la secuela:

- a) Velocidad de operación;
- b) Tiempo de reparación;
- c) Costo de producción;
- d) Confiabilidad individual y;
- e) Capacidad de producción

Así, la presente investigación trata de la asignación de redundancias considerando que al revés de componentes simples considerase subsistemas complejos con atributos distintos de las triviales (masa, volumen, confiabilidad) encontrados en las resoluciones de este tipo de problema combinatorio.

4. Diseño metodológico

4.1 Definición de los modelos conceptual y de simulación del proceso de producción:

El sistema automatizado adoptado es representado pelo caso idealizado en cuestión y es de tipo a eventos discretos. Este tipo de sistema requiere para su tratamiento lenguajes y herramientas de modelaje específicas. En esta investigación adoptaran se las red de Petri para la especificación del modelo conceptual del proceso de producción.

El modelo conceptual Petri permitió el planteamiento de todas las operaciones involucradas, estableciendo claramente las transiciones y operaciones del sistema. Permitted también de manera facilitada y visual cambios en el modelo de simulación, así como la jerarquización de las secuencias de operaciones.

La ventaja de esta metodología delante las diversas lenguajes existentes es que su formalismo basase en una modelaje grafica simplificada con pocas reglas sintácticas [117-120]. Otra característica de interese es que el desarrollo lógico de un modelo Petri se asemeja a lógica de programación orientada a bloques de los entornos de simulación actuales como: ARENA, PROMODEL e o SIMUL8.

El modelo conceptual norteó la construcción del modelo de simulación lo cual en su forma final contiene cerca de 500 variables y:

9 subsistemas correspondientes a cada una de las operaciones del modelo, de acuerdo con el ítem 1.2, totalizando - 45 módulos básico de proceso; 135 módulos de decisión; 90 módulos de asignación de valores a las variables del proceso.

9 módulos de asignación de valores de las variables globales del sistema que son: el número total de fallas del sistema; tiempo total de paralización del sistema; disponibilidad total del sistema; índice de pérdida de calidad del producto; los costos globales (CMODC, CPPROS, CGM, CMANS) y sus correspondientes a nivel de subsistemas ($cmodcij$, $cpppros_{ij}$, cgm_{ij} y $cmans_{ij}$)

Vale resaltar todavía la característica estocástica del modelo de simulación. Eso está en el fundamento del argumento principal de esta investigación debido la imposibilidad de la construcción de una función que represente todas las relaciones del sistema. Destacase también en el modelo de simulación los diversos aspectos aleatorios que concurren en el proceso simulado como, por ejemplo: el proceso de falla y reparación de acuerdo con la tabla 4; la degradación de la velocidad de producción de una máquina a partir de la ocurrencia de un determinado número de fallas, además del proceso de colas producidas en las máquinas regido pelas variables: tiempo de operación de la máquina, tiempo de espera por una pieza, tiempo de espera por otra operación y el tiempo de falla.

Ya los modelos estocásticos de falla/reparación son atribuidos distintamente a la cada máquina/robot, tabla 4. En lo que concierne a las fallas/reparación de cada máquina/robot fueran distribuidos exponencialmente con promedios distintos, tabla 4. En esta tabla se enseña también las distribuciones de los tiempos de inspección de mantenimiento preventivo programados. Estos tiempos varían de acuerdo con el tipo de inspección que en su vez dependen del número de piezas producidas.

Es importante observar que tanto la topología cuanto la asignación de valores en el modelo de simulación es una tarea mucho dependiente de la experiencia del modelador. Como consecuencia, la descripción más pormenorizada del modelo de simulación de esta investigación sería además de largo también impropio. En el anexo 1 se presentan el diagrama Petri con las tablas de definición de las operaciones y transiciones del sistema.

Tabla 4 Distribuciones de falla y reparación y los tiempos de inspección

Fallas	Tipo	Distribución de falla	unidades	Basado en tiempo	Tiempo de reparación	unidades
Falla 1	Tiempo	EXPO(350)	Horas	EXPO(350)	EXPO(1.5)	Horas
Falla 2	Tiempo	EXPO(400)	Horas	EXPO(400)	EXPO(1.0)	Horas
Falla 3	Tiempo	EXPO(450)	Horas	EXPO(450)	EXPO(0.85)	Horas
Falla 4	Tiempo	EXPO(550)	Horas	EXPO(550)	EXPO(0.65)	Horas
Falla 5	Tiempo	EXPO(650)	Horas	EXPO(650)	EXPO(0.50)	Horas
Falla inspección 1	6 Basado en tiempo	1.0	Horas	100000	EXPO(0.25)	Horas
Falla inspección 2	7 Basado en tiempo	1.0	Horas	150000	EXPO(0.50)	Horas
Falla inspección 3	8 Basado en tiempo	1.0	Horas	250000	EXPO(1.0)	Horas
Falla inspección 4	9 Basado en tiempo	1.0	Horas	300000	EXPO(0.65)	Horas
Falla inspección 5	10 Basado en tiempo	1.0	Horas	350000	EXPO(0.85)	Horas
Falla 11	Tiempo	EXPO(650)	Horas	EXPO(650)	EXPO(1.25)	Horas
Falla 12	Tiempo	EXPO(850)	Horas	EXPO(850)	EXPO(1.0)	Horas
Falla 13	Tiempo	EXPO(1050)	Horas	EXPO(1050)	EXPO(0.75)	Horas
Falla 14	Tiempo	EXPO(1250)	Horas	EXPO(1250)	EXPO(0.50)	Horas
Falla 15	Tiempo	EXPO(1450)	Horas	EXPO(1450)	EXPO(0.25)	Horas
Falla 16	Tiempo	EXPO(850)	Horas	EXPO(850)	EXPO(1.25)	Horas
Falla 17	Tiempo	EXPO(1050)	Horas	EXPO(1050)	EXPO(1.0)	Horas
Falla 18	Tiempo	EXPO(1250)	Horas	EXPO(1250)	EXPO(0.75)	Horas
Falla 19	Tiempo	EXPO(1450)	Horas	EXPO(1450)	EXPO(0.50)	Horas
Falla 20	Tiempo	EXPO(1650)	Horas	EXPO(1650)	EXPO(0.25)	Horas
Falla 21	Tiempo	EXPO(300)	Horas	EXPO(300)	EXPO(1.5)	Horas
Falla 22	Tiempo	EXPO(350)	Horas	EXPO(350)	EXPO(1.0)	Horas
Falla 23	Tiempo	EXPO(400)	Horas	EXPO(400)	EXPO(0.85)	Horas
Falla 24	Tiempo	EXPO(450)	Horas	EXPO(450)	EXPO(0.75)	Horas
Falla 25	Tiempo	EXPO(500)	Horas	EXPO(500)	EXPO(0.50)	Horas
Falla 26	Tiempo	EXPO(500)	Horas	EXPO(500)	EXPO(1.5)	Horas
Falla 27	Tiempo	EXPO(550)	Horas	EXPO(550)	EXPO(1.0)	Horas
Falla 28	Tiempo	EXPO(600)	Horas	EXPO(600)	EXPO(0.85)	Horas
Falla 29	Tiempo	EXPO(650)	Horas	EXPO(650)	EXPO(0.65)	Horas
Falla 30	Tiempo	EXPO(700)	Horas	EXPO(700)	EXPO(0.50)	Horas

4.2 Formulación de las variables, objetivos y restricciones del modelo completo (simulación/optimización)

Las variables del proceso de simulación optimización pueden ser clasificadas como exógenas y endógenas al código principal, o sea, al algoritmo genético elitista AGE. Las variables exógenas son aquellas procesadas en el interior del simulador a partir de un determinado escenario de operación. Los elementos de cada escenario (configuración, mantenedores, costos y confiabilidades) son representados en el modelo, por un conjunto de variables endógenas generadas en el AGE.

Los objetivos de optimización, en su vez, son calculados con contribución de los dos tipos de variables. Para el caso idealizado en cuestión, son utilizados 4 objetivos en conflicto, los cuales deben ser optimizados concomitantemente en la secuela:

Costo total de mantenimiento del sistema (minimizar)

Confiabilidad total del sistema (maximizar)

Costo total de operación del sistema (minimizar)

Número de mantenedores (minimizar)

El cálculo de estos 4 objetivos está condicionado a uno respectivo escenario de operación. Cada escenario, o sea, cada individuo del AGE es generado aleatoriamente y constituido por las variables endógenas en la secuela:

- Número de mantenedores – variable entera $numan \in \mathbb{Z}$
- Asignación de máquinas/robots – variable binaria $xbin_{ij}$
- Confiabilidades esperadas de cada máquina asignadas o no en el sistema – variable real entre 0 y 1;
- Las 4 componentes del costo total de operación del sistema – variables reales
- *cmapa* – Costo de la materia prima aplicada en el sistema
- *cmoop* - Costo de la mano de obra para la operación del sistema
- *cenerg* Costo de la electricidad y otros combustibles
- *cenalt* – Costo del empaque, almacenamiento y transporte ,
- *CFTM* – Costo fijo total de mantenimiento – variable real

Es importante tener en cuenta que algunas de estas variables hacen parte de los objetivos de optimización o mismo representan el propio objetivo de optimización, como en el caso de las variables *CFTM* y *numan* respectivamente. Vale todavía destacar que la variable $xbin_{ij}$ representa la asignación de una maquina/robot *i* en la etapa *j* del sistema. Esta variable juega un papel primordial en la producción de la variabilidad de los escenarios de operación, pudiendo ser declarada según:

$$\begin{aligned}
 & xbin_{ij} \\
 & 0, \text{ para máquina robot no asignado} \\
 & 1, \text{ para } \frac{\text{máquina}}{\text{robot}} \text{ asignado}
 \end{aligned} \tag{1}$$

La tarea de optimizar la confiabilidad de uno sistema siempre implica en el consumo de recursos, sean ellos humanos, estructurales o financieros. Tales recursos están representados en el modelo por otro subgrupo de variables exógenas, las cuales representan costos directamente asociados a la asignación de redundancias en la secuela:

$cmtt_{ij}$ – costo de utilización de herramientas, partes y mano de obra para mantenimiento de la máquina *j* en el subsistema *i*.

$cinst_{ij}$ – costo de instalación de una maquina *j* en el subsistema *i*.

$cppm_{ij}$ - costo por la pérdida de producción de la máquina *j* en el subsistema *i* debido a una parada por ocurrencia de una falla.

Costo general de administración (*CGA*) del mantenimiento correspondiendo 10% de la suma de los costos $cmtt_{ij}$, $cinst_{ij}$ e $cppm_{ij}$.

$$CGA = \sum_{i=1}^k \sum_{j=1}^n xbin_{ij} * cinst_{ij} + cmtt_{ij} + cppm_{ij} \tag{1}$$

Estos costos hacen parte de las dos funciones de costo principales del modelo: el costo total de mantenimiento y el costo total de operación del sistema. Tales funciones acaban en su vez se convirtiendo en dos de los objetivos de optimización (objetivos 1 y 3).

El costo total de mantenimiento (*CMANS*) posee dos componentes: el costo fijo total de mantenimiento *CFTM* (variable endógena) el costo variable total de mantenimiento (*CVTM*) (variable exógena). En la expresión del costo variable total de mantenimiento entran todavía más tres variables reales exógenas:

CMODC – Variable costo/hora de la mano de obra directa

CPPROS – Costo por la pérdida de producción debido a no operación
DTS - Disponibilidad total del sistema

La variable *CMODC* es exponencialmente dependiente del tiempo total de paralización, termo $numan * e^{\frac{1}{70} * tpm_{ij}}$ y dada por:

$$CMODC = \sum_{i=1}^n \sum_{j=1}^m (numan * e^{\frac{1}{70} * tpm_{ij}} + to_{ij} + tpm_{ij}) * cmodc_{ij} * xbin_{ij} \quad (2)$$

Dónde:

tpm_{ij} - tiempo de paralización de máquina *i* en el subsistema *j*;

$cmodc_{ij}$ Costo de la mano de obra especializada por el mantenimiento de la máquina *i* en el subsistema *j*;

to_{ij} - Tiempo de operación de la máquina *i* en el subsistema *j*;

Cada vez que una máquina/robot entra en el periodo de no operación resultante de una falla, uno costo estimado es asociado. Así, el costo total por la pérdida de producción del sistema *CPPROS* es una variable real exógena definida por las expresiones

$$cppm_{ij} = com_{ij} \times \frac{ppm_{ij}}{npepm_{ij}} \times tr_{ij} * xbin_{ij} \quad (3)$$

$$CPPROS = \sum_i^n \sum_j^m cppm_{ij} \quad (4)$$

Onde:

- com_{ij} costo de operación de la maquina *i* en el subsistema *j*
- ppm_{ij} productos producidos por la máquina *i* en el subsistema *j*
- $npepm_{ij}$ número de productos esperados producidos sin falla por la máquina *i* en el subsistema *j*
- tr_{ij} tiempo de reparación de la máquina *i* en el subsistema *j*

Por lo tanto el costo variable total de mantenimiento, *CVTM* (variable real) es representado por la expresión:

$$CVTM = \frac{\sum_{i=1}^n \sum_{j=1}^m xbin_{ij} * numan * e^{\frac{1}{70} * tpm_{ij}} + to_{ij} + tpm_{ij} * cmodc_{ij} * xbin_{ij} + 1.1 * \sum_{i=1}^k \sum_{j=1}^n xbin_{ij} * cinst_{ij} + cmtt_{ij} + cppm_{ij}}{DTS} \quad (5)$$

La composición final de la función del costo total de mantenimiento del sistema, $CTMS$ es dada por:

$$CTMS = CFTM + \frac{CMODC + 1.1 * CGA}{DTS} \quad (6)$$

El costo total de operación del sistema $CTOP$ (variable real) es el resultado de la suma de los costos relacionados con la aplicación de recursos e insumos expresos por las variables endógenas reales $cmapa$, $cmoop$, $cenerg$ y $cenalt$ con limites pre-establecidos, y el costo variable total de mantenimiento

$$TOC = TPPS * 0,05 * cmapa + TPPS * 0,15 * cmoop + TPPS * 0,10 * cenerg + TPPS * 0,06 * cenalt + CVTM \quad (7)$$

Donde

- $cmapa$ – Costo de la materia prima aplicada en el sistema
- $cmoop$ - Costo de la mano de obra para la operación del sistema
- $cenerg$ Costo de la electricidad y otros combustibles
- $cenalt$ – Costo del empaque, almacenamiento y transporte
- $TPPS$ - Número total de productos producidos por el sistema

Finalmente, la confiabilidad total del sistema (R_s) es una variable real limitada entre 0 – 1 y fue calculada de acuerdo con la estructura del sistema, paralelo/serie, con posibilidad de asignación de máquinas/robots redundantes no idénticos:

$$R_s = \prod_{i=1}^k 1 - \prod_{j=1}^{n_j} (1 - r_{ij} * xbin_{ij}) \quad (8)$$

Además de las variables descritas anteriormente, el proceso de simulación suministra otras variables exógenas estocásticas que dependen exclusivamente del modelo de simulación y que, por lo tanto, dependen del desempeño del sistema para uno determinado escenario de operación:

$NTFS$ – número total de fallas del sistema;

$TPBPS$ – número total de productos buenos producidos por el sistema. Es una variable que identifica los productos que fueran producidos si ninguna alteración o considerados buenos.

DTS – disponibilidad total del sistema

Así, la tarea de evaluar la adecuación de uno escenario de operación (individuo) ha uso, directo o indirecto, de ambos los tipos de variables del proceso (endógenas y exógenas). En esto sentido cabe a las restricciones del problema la fundamental tarea de balizar el proceso de determinación de la adecuación de cada solución/escenario de operación.

En el proceso de evaluación verificase la pertinencia o no del valor de las variables de las restricciones a los límites predeterminados por el analista (**LI** límite inferior – **LS** – límite superior). Las doce restricciones citadas, existe cuatro relacionadas a los costos. Estos costos tienen su valor directamente relacionado al proceso de asignación de redundancias, siendo totalizados en el simulador a través de variables exógenas globales identificadas por letras mayúsculas así definidas:

- Costo total de utilización de herramientas, partes y mano de obra para el mantenimiento de las máquinas (variable exógena)

$$CMTTPM = \sum_{i=1}^n \sum_{j=1}^m cmttpm_{ij} * xbin_{ij} \leq LS \quad (9)$$

- Costo total de instalación de máquinas (variable exógena)

$$CINST = \sum_{i=1}^n \sum_{j=1}^m cinst_{ij} * xbin_{ij} \leq LS \quad (10)$$

- Costo total pérdida de producción de las máquinas debido a una paralización por ocurrencia de una falla (variable exógena)

$$CPPM = \sum_{i=1}^n \sum_{j=1}^m cppm_{ij} * xbin_{ij} \leq LS \quad (11)$$

- Costo general de administración del mantenimiento correspondiendo a 10% de la suma de los costos $cmttpm_{ij}$, $cinst_{ij}$ e $cppm_{ij}$ (variable exógena)

$$CGA = \sum_{i=1}^n \sum_{j=1}^m (cmttpm_{ij} + cinst_{ij} + cppm_{ij}) * xbin_{ij} \leq LS \quad (12)$$

El segundo conjunto de restricciones está relacionado con las variables de los objetivos principales siendo así definidas:

- Confiabilidad total del sistema (Variable exógena)
 $LI \leq R_s \leq LS$
- Costo total de operación del sistema (Composición de variables – endógenas y exógenas)
 $LI \leq CTOS \leq LS$
- Costo total de mantenimiento del sistema (Composición de variables – endógenas y exógenas)
 $LI \leq CTMS \leq LS$
- Número de mantenedores (Variable endógena)
 $LI \leq numan \leq LS$
- Costo fijo total de mantenimiento del sistema (Variable endógena)
 $LI \leq CFTM \leq LS$

El tercer y último conjunto de restricciones está relacionado con las tres variables exógenas estocásticas oriundas exclusivamente del proceso de simulación y ya mencionadas:

- Tiempo total de paralización del sistema $LI \leq TPS \leq LS$
- Número total de fallas del sistema $LI \leq NTFS \leq LS$
- Disponibilidad total del sistema $LI \leq DTS \leq LS$

De esta manera el funcionamiento de todo el proceso está basado en la correcta definición de los límites de las restricciones. Tal tarea depende mucho del grado de familiaridad del analista con el sistema, pues, sin este conocimiento difícilmente se establecerán los límites adecuados al proceso. En otras palabras los límites representan los horizontes de los espacios de soluciones y de variables del problema que mantendrán el proceso de búsqueda en la dirección deseada.

4.3 El proceso de interacción entre el simulador y el optimizador

Diferentemente del proceso de optimización tradicional basado en la construcción de funciones objetivo, se tiene aquí un proceso de optimización que no las utiliza de manera usual. Todo el proceso es representado por uno conjunto de variables que son definidas tanto en el AGE como en el simulador. Algunas son incorporadas a las restricciones y otras se convierten en los propios objetivos de optimización como se podrá ver en la secuela.

El proceso de simulación optimización aquí empleado haz uso de dos herramientas computacionales, el entorno de simulación ARENA (simulador) y el algoritmo genético NSGA II (optimizador). El simulador produce las variables que serán utilizadas por el AGE en la determinación de la adecuación de la solución del problema.

La interfaz entre estas herramientas constituyese básicamente en una función insertada en el código del AGE, que coordina el proceso de lectura y cambio de variables endógenas y exógenas entre los programas. Todo el proceso simulacional ocurre de forma única, o sea, coordinado dentro de la rutina principal del AGE. El modelo de simulación entonces desempeña un papel importante en la tarea de verificación de la adecuación del escenario de operación, pues, contribuye con las variables que representan la dinámica y la aleatoriedad del proceso.

El simulador genera subsidios parciales para la evaluación del escenario, o sea, él no debe ser entendido como una función de adecuación solamente. El criterio de parada introducido en el modelo de simulación es el tiempo de funcionamiento del sistema automatizado, o sea, el número horas/días/meses que el sistema debe funcionar durante la simulación.

La posición del simulador en el flujo grama general del optimizador (NSGA II) es de fundamental importancia para el proceso de simulación empleado. Como enseñado en la figura 8, cabe a lo AGE producir secuencialmente los escenarios de operación. El simulador recibe un escenario de cada vez y simula su operación, tan realísticamente cuanto más detallado sea su respectivo modelo. Cabe a lo AGE, en la etapa posterior, evaluar la adecuación del escenario según el criterio de no dominancia mencionado anteriormente.

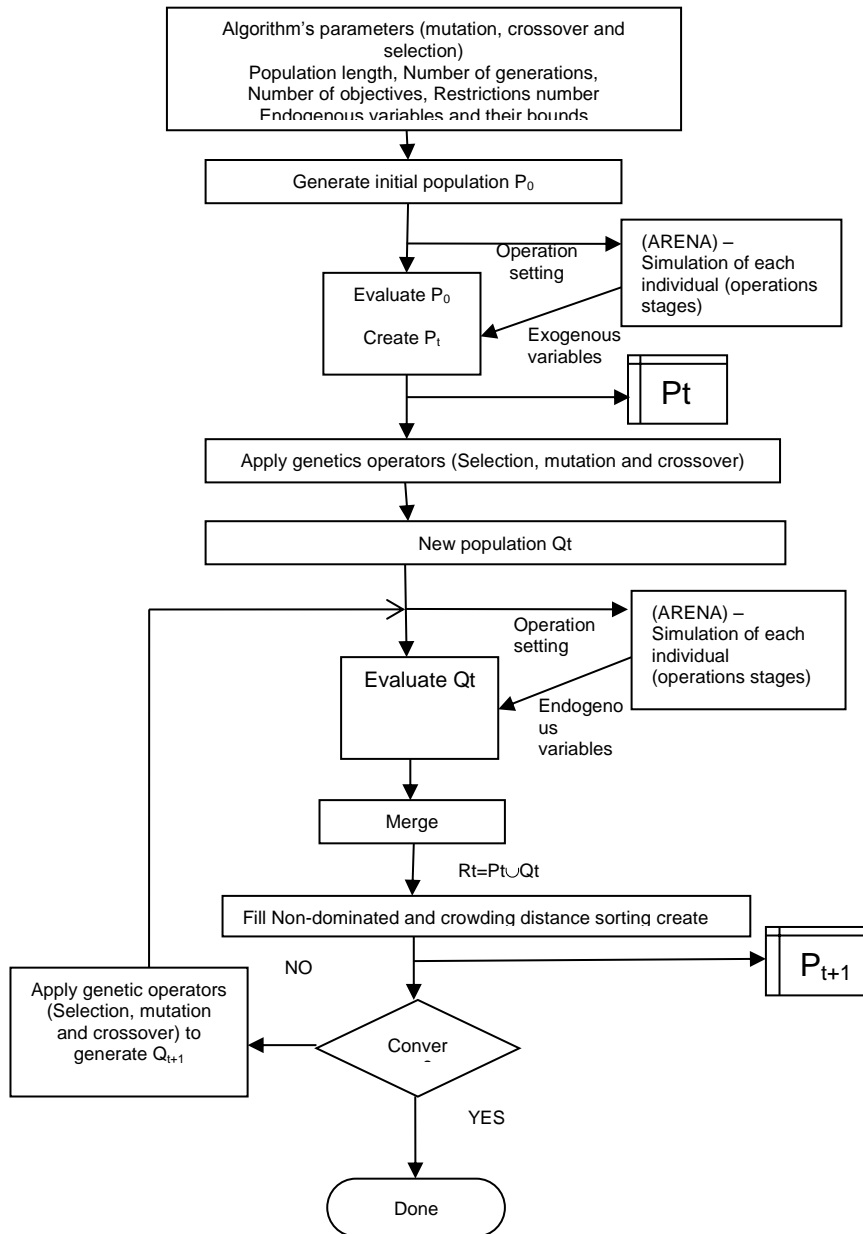


Figura 4 Flujo grama de la estructura simulación/optimización

Cuando un escenario de operación es creado y enviado al simulador, archivo entradaArena.txt, se establece la configuración de las máquinas que serán accionadas en el proceso, y a medida que el sistema avanza en el tiempo son calculadas las variables exógenas del sistema: R_s , CGA , $CINST$, $CMTTPM$, $CMODC$, $NFTS$, $TPPS$, $TPBPS$, DTS , $CPPM$ y $el CVTM$.

Como visto anteriormente parte de estas variables entra en el cálculo de los objetivos que serán optimizados, otras caracterizan aspectos de la eficiencia del sistema para la configuración de máquinas establecida en el escenario de operación. Todas, sin

embargo, sea directa o indirectamente, son incorporadas a la rutina del AGE de verificación de las restricciones del problema.

En esta rutina, el valor de las variables exógenas, así como de las variables endógenas (supra-definidas), es comparado, en las restricciones del problema, a límites pre-establecidos por el analista. Es justamente la satisfacción de estos límites que determina a adecuación del escenario en cuestión. El escenario será desechado en el caso de una o más de estas restricciones sea violada y, acepto, casa hay satisfecho a todas las restricciones, así como a todos los objetivos de optimización.

Determinada la adecuación del escenario, el proceso de simulación reanudase con un nuevo individuo/escenario de operación (generado anteriormente en el AGE). El criterio de parada global del proceso es el número de generaciones, donde cada generación es constituida por un conjunto de individuos/escenarios de operación. Vale resaltar que el AGE manipula una generación de cada vez, en cuanto que el simulador manipula apenas un individuo/escenario de operación a la vez. Esta característica demanda la inclusión de un mecanismo de espera en el AGE, permitiendo al simulador realizar la simulación de todos los individuos/escenarios de operación.

Para facilitar la lectura de los datos de salida del simulador fue generado un archivo llamado *saidaArena_g*"X".txt para $1 \leq X \leq N$ dónde N es el número total de generaciones, los datos de cada archivo *saidaArena.txt* son escritos en forma de línea no en columna como fueran originalmente hechos en los archivos *saidaArena_g*"x"*_i*"y".txt así vamos tener para N=50 cincuenta archivos contiendo 20 líneas cada que corresponde a cada uno de los veinte individuos de la generación de los experimentos, 2 a 4, de 50 generaciones.

Al final de todo el proceso evolutivo el archivo, generado por el AGE, **best_pop.out** contiene las mejores soluciones de las "N" generaciones elegidas anteriormente. Las soluciones son presentadas en forma de una tabla donde las tres primeras columnas presentan los mejores resultados para cada uno de los "M" objetivos del problema. En la secuela viene las "z" columnas correspondientes al resultado obtenido en el cálculo de cada una de las restricciones, después vienen las columnas de las variables reales "xreal" del problema, después las columnas de las variables binarias "xbin" del problema y las tres columnas finales que corresponden a la violación de las restricciones, rango y "*crowding distance*".

Cada archivo *entradaArena.txt* que corresponde a un escenario de operación es salvo en la forma *entradaArena_g"x"_i"y".txt* donde el "g" corresponde al número de la generación y el "y" al número del escenario.

4.4 Elección del mejor escenario de operación

Tratándose de optimización multiobjetivo, lo que se espera de los algoritmos empleados en esta clase de problemas es que presenten un conjunto de soluciones que estén comprobadas con todos los objetivos. Sin embargo, la elección final de una solución, dado que no se tiene una única solución y si un conjunto de ellas, se da por el empleo de un procedimiento de agregación de información que puede ser realizado *a priori*, *mientras* o *a posteriori* del proceso.

En *a priori* los criterios son establecidos antes del proceso ser ejecutado y nortearan la construcción del conjunto óptimo de Pareto.

En *mientras* los criterios son insertados mientras el desarrollo del proceso que, pueden ser variables resultantes de iteraciones anteriores o insertadas por el usuario.

Y, por fin en *a posteriori* el proceso se desarrolla sin interferencia del tomador de decisiones y al final sobre el conjunto de Pareto encontrado se agrega la información necesaria para la selección de una solución.

Un procedimiento de agregación de información puede ser de naturaleza técnica o cuantitativa y no técnica o cualitativa. La primera se relaciona con datos técnicos como especificaciones y variables de proceso relacionadas con el ajuste del sistema como un todo. La segunda expresa la opinión del personal involucrado con el proceso de decisión.

En esta investigación optase por el proceso a posteriori dado que al término de todos los experimentos los conjuntos de Pareto resultantes fueran preparados para la aplicación de los criterios de elección, tablas 7-12

4.5 Declaración del modelo de prueba no lineal, mono-objetivo y no estocástico

Adicionalmente al problema multiobjetivo pretendiese también producir una simplificación para su resolución. Tratase del desarrollo de un modelo mono objetivo determinista y estático sin la influencia de las variables estocásticas, para esto un modelo basado en la abordaje ϵ restricción (*ϵ -constraint methods or approach*) Deb [79].

Con los resultados del modelo simplificado se hará una comparación con la estructura de simulación/optimización.

Entonces, el problema multiobjetivo original, descrito anteriormente, con cuatro objetivos fue convertido en un modelo mono-objetivo no estocástico. Fueran fijados los índices de confiabilidad individuales de cada máquina.

Los valores de los costos de utilización de herramientas, partes y mano de obra para el mantenimiento de la máquina j en el sub-sistema i ($CMTTPM_{ij}$), del costo de instalación de una máquina j en el subsistema i ($CINST_{ij}$), del costo asociado a la pérdida de producción de máquina j en el subsistema i debido a una paralización por ocurrencia de una falla ($CPPM_{ij}$), fueran prefijados y suministrados tanto cuanto la cantidad de productos producidos ($Tpps$) para cada tiempo de funcionamiento del sistema, 720, 1440 y 2160 horas. No fueran consideradas las tasas de fallas de las máquinas como también las tasas de reparación de las mismas. La disponibilidad total del sistema (DTS) fue considerada 0.90 para cada uno de los tiempos de funcionamiento simulados. El modelo garantiza que por lo menos habrá una máquina asignada en cada una de las etapas o subsistemas.

El segundo y tercero objetivos de optimización, que representan el costo total de mantenimiento del sistema y el costo total de operación respectivamente, fueran convertidos en restricciones del problema. Y el cuarto objetivo siguió siendo una variable ***xreal 0***.

La constante A que aparece dentro de la ecuación de la primera restricción es un valor fijo que representa el costo $CMODC$ que, en la versión multiobjetivo, es una variable aleatoria que depende del tiempo de paralización del sistema. Así el valor de la *constante* A , que representa un valor fijo del costo:

$$CMODC = \sum_{i=1}^n \sum_{j=1}^m (n_{uman} * e^{\frac{1}{70} * tpm_{ij}} + t_{o_{ij}} + tpm_{ij}) * cmodc_{ij} * xbin_{ij} \quad (14)$$

Y tiene su valor de 3973,99, 7959,12 y 11989,40 (U.M.) para 720, 1440 y 2160 horas respectivamente.

El modelo fue programado en lenguaje AMPL e fue utilizado o “***solver***” KNITRO 6.0 versión estudiantil debido al número reducido de variables empleado. Fueran utilizadas 45 variables binarias y 5 variables enteras con 17 restricciones lineales. Así la declaración del problema mono-objetivo se presenta en la secuela:

Datos:

e , conjunto de etapas o subsistemas;

m , conjunto de máquinas que podrán actuar en el sistema

n , número máximo de máquinas;

k , número máximo de subsistemas o etapas

$C_{mtpm_{ij}}$ – costo de utilización de herramientas, partes y mano de obra para mantenimiento de la máquina j en el subsistema i .

$C_{inst_{ij}}$ – costo de instalación de una maquina j en el subsistema i .

$C_{ppml_{ij}}$ - costo por la pérdida de producción de la máquina j en el subsistema i debido a una parada por ocurrencia de una falla.

$tpps$, número máximo de productos producidos por el sistema.

DTS , disponibilidad total del sistema

Y

r_{ij} - índices de confiabilidad de la máquina j en el subsistema i

Define:**Variable de asignación de maquinas**

$xbin_{ij}$ – $i \in e, j \in m$ – variable de asignación de asume:

$$\left\{ \begin{array}{l} 1, \text{ para máquina asignada} \\ 0, \text{ para máquina no asignada} \end{array} \right. \quad \mathbf{xbin[0].....xbin[49]}$$

$5 \leq xreal\ 0 \leq 20$ **Variable número de mantenedores del sistema**

$0.1 \leq xreal\ 1 \leq 1.0$ – $cmapa$ –

$1.0 \leq xreal\ 2 \leq 3.0$ – $ccoop$ –

$1.0 \leq xreal\ 3 \leq 7.0$ – $cenerg$

$1.0 \leq xreal\ 4 \leq 5.0$ – $cenalt$

Variable costo fijo total de mantenimiento del sistema considerando la configuración óptima del sistema

$$250 \times 10^3 \leq xreal\ 5 \leq 350 * 10^3$$

Maximizar

$$R_s = \prod_{i=1}^k \prod_{j=1}^{n_j} (1 - r_{ij} * xbin_{ij})$$

Sujeto a:

$$CTMS = xreal\ 5 + \frac{xreal\ 0 * CONSTANTE\ A + 1.1 * \sum_{i=1}^k \sum_{j=1}^{n_j} xbin_{ij} * cinst_{ij} + cmttpm_{ij} + cppm_{ij}}{DTS} \leq 3500 \times 10^3$$

$$TOC = tpps * \sum_{k\ n} 0,05 * xreal[1] + 0,15 * xreal[2] + 0,10 * xreal[3] + 0,06 * xreal[4] + CVTM \leq 8000 * 10^3$$

$$xbin_{ij} \leq n$$

$$CPPROS = \sum_{i\ j} \sum_{k\ n} (cppm_{ij} * xbin_{ij}) \leq 900 \times 10^3$$

$$CINST = \sum_{i=1}^k \sum_{j=1}^{n_j} cinst_{ij} * xbin_{ij} \leq 1900 * 10^3$$

$$CPPM = \sum_{i=1}^k \sum_{j=1}^{n_j} cppm_{ij} * xbin_{ij} \leq 10 * 10^3$$

$$CGA = \sum_{i=1}^k \sum_{j=1}^{n_j} xbin_{ij} * cinst_{ij} + cmttpm_{ij} + cppm_{ij} \leq 260 * 10^3$$

5. Experimentos

Fueran realizados 6 experimentos al todo:

- **Experimentos 1, 2 y 3** mono-objetivo constituido por 3 corridas para 720, 1440 y 2160 horas simuladas
- **Experimentos 4, 5 y 6** de 720, 1440 y 2160 horas simuladas con 50 generaciones de 20 individuos(1000 individuos generados por corrida) respectivamente
- **Experimento 7** de 720 horas simuladas con 100 generaciones de 20 individuos (2000 individuos generados)
- **Experimento 8** de 720 horas simuladas con 150 generaciones de 20 individuos (3000 individuos)

Cabe recordar que cada individuo corresponde a un escenario de operación y que cada conjunto de 720 horas corresponde a un mes de operación sin interrupción del sistema. Estos individuos fueran sometidos al proceso de evolución del AGE lo cual, de acuerdo con el ya discutido, es guiado por los operadores de reproducción. Las probabilidades de ocurrencia de estos operadores fueran:

- 0.7 para el cruzamiento y 0,02 para la mutación – para las variables reales;
- 0.7 para el cruzamiento y 0.022 para la mutación – para las variables binarias;

Estos operadores, que son responsables por la amplitud y diversidad de la búsqueda, tuvieron sus probabilidades de ocurrencia mantenidas constantes para garantizar una cierta homogeneidad de búsqueda entre los experimentos multiobjetivos. De esta manera, se evitaran oscilaciones muy grandes dos resultados entre los 5 experimentos multiobjetivos.

Los experimentos produjeran una grande cantidad de datos agrupados por generaciones. Cada generación tuvo calculada su promedio, desviación estándar y variancia.

6. Resultados: análisis y discusión

El análisis de los resultados organizase en tres sub ítems: el primero presenta los resultados para los objetivos principales del experimento mono objetivo; el segundo haz una larga discusión acerca de la convergencia del AGE; el tercer discute y compara los resultados obtenidos en todos los experimentos. En este último buscase también responder la pregunta del final del ítem 1.2:

¿Cuál el escenario óptimo de operación para el caso idealizado en cuestión, o sea, aquello que maximiza la confiabilidad total del sistema y concomitantemente minimiza o empelo de mantenedores y los costos de operación y mantenimiento?

En todos los experimentos son evaluados 4 objetivos. En esta evaluación haz uso de índices de confiabilidad, costos y el número de mantenedores. Los índices de confiabilidad varían entre 0 y 1. Ya costos y el número de mantenedores pueden asumir valores cualesquiera, a pesar de los mantenedores sean una variable entera. De manera a poder comparar estos objetivos de naturaleza distinta, son necesarios ajustes. Por lo tanto, adoptase el procedimiento de normalización en la secuela:

$$x_n = \frac{x_i}{\sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}} \quad (15)$$

Donde

x_n – valor de un punto normalizado

x_i - valor de um individuo promedio de la generación i con $i=1, \dots, n$

n – número total de generaciones

N – orden de la matriz

6.1 Experimento 1 mono objetivo

Una vez realizadas las simulaciones los resultados obtenidos fueran para los objetivos:

Objetivo 1 – Minimización del costo total de mantenimiento del sistema (CTMS)

Objetivo 2 – Maximización del confiabilidad del sistema R_s

Objetivo 3 – Minimización del costo total de operación del sistema (CTOS)

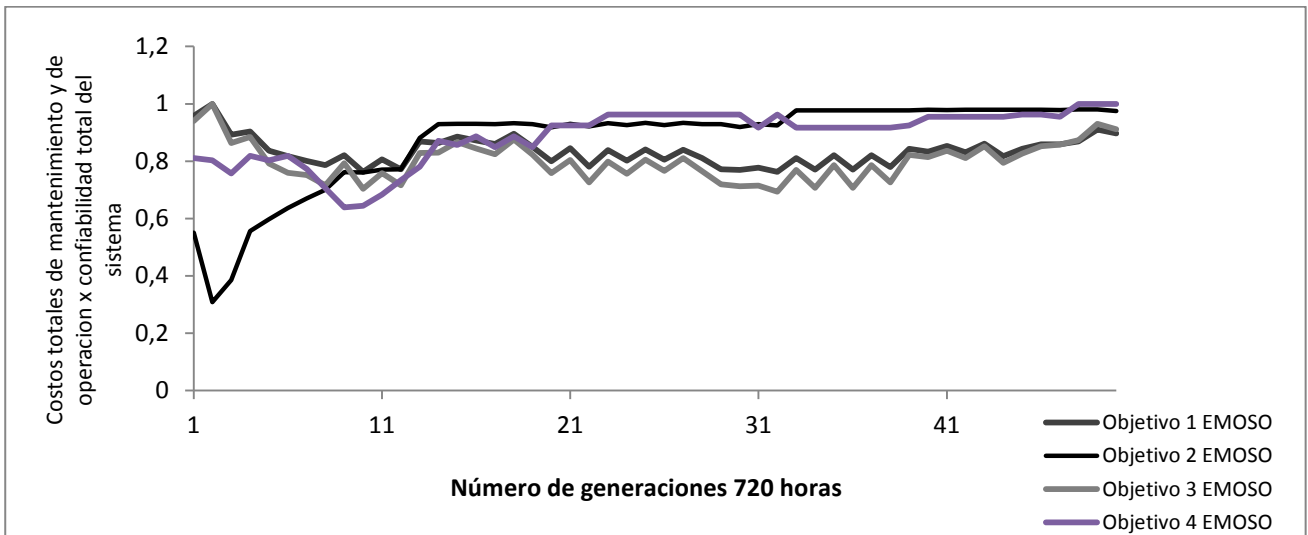
Objetivo 4 – Minimización del número de mantenedores del sistema

Tabla 5 Resultados de los experimentos (1, 2 y 3) mono objetivo

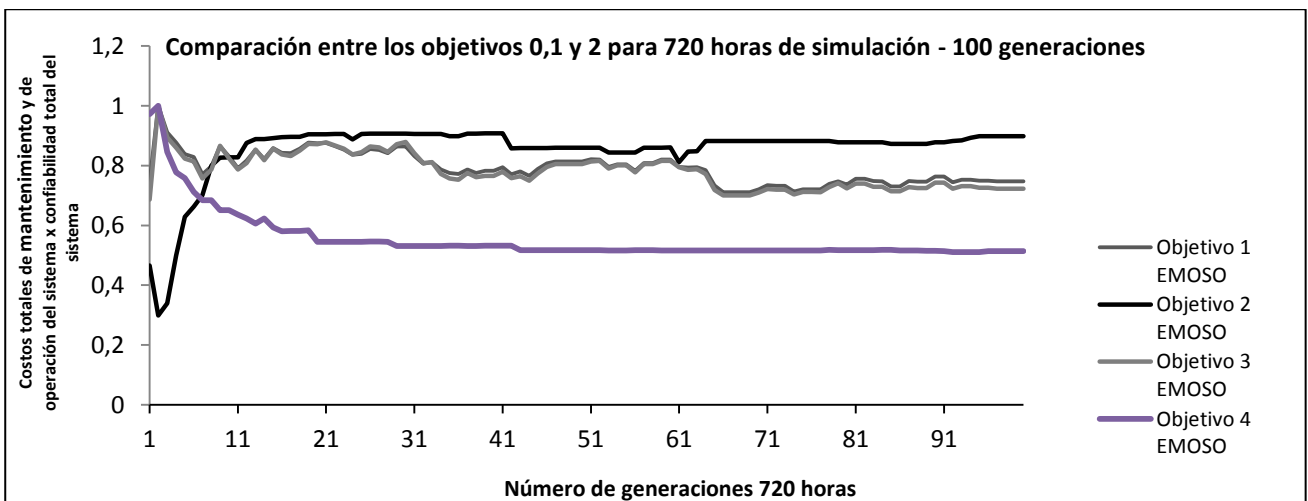
Horas simuladas	Objetivo 1	Objetivo 2	Objetivo 3	Objetivo 4	Asignación
720 horas	2522403,22	0,888	4123482	6	18
1440 horas	2700065,36	0,888	5902222	6	18
2160 horas	2930878,79	0,787	7734114	7	17

6.2 Estudios de convergencia

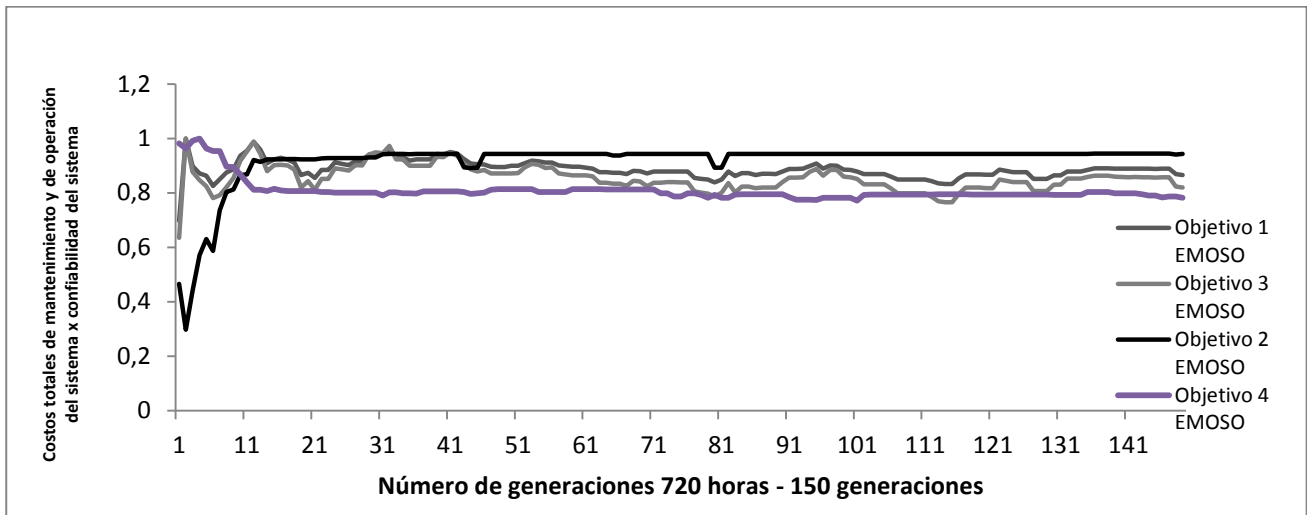
En los experimentos multiobjetivos realizados observase que existe una tendencia a la disminución de la desviación estándar y variancia de la generaciones con la evolución del proceso y a medida que el AGE acercase de su criterio de parada. Como será discutido en la secuela en AGE puede ser caracterizado, entonces, como una secuencia de eventos discretos independientes, o sea, una cadena de Markov. De esta manera el proceso tiene convergencia asegurada. Las gráficas 2-6 enseñan esta convergencia del proceso para todos los 4 objetivos para un mismo tiempo de simulación (720 horas), pero, con el número de generaciones variable, respectivamente 50, 100 y 150 generaciones. Las gráficas presentan los promedios de los objetivos en cada generación.



Gráfica 2 Comparación entre los objetivos 1, 2 ,3 y 4 720 horas simuladas – 50 generaciones



Gráfica 3 Comparación entre los objetivos 1, 2,3, y 4 para 720 horas simuladas - 100 generaciones



Gráfica 4 Comparación entre los objetivos 1,2, 3 y 4 para 150 generaciones y 720 horas simuladas - 150 generaciones

Se evidenció también que a pesar de la tendencia de disminución de la dispersión, el algoritmo sigue oscilando indicando que su proceso de búsqueda continúa hasta que encuentre el número de generaciones especificado. La convergencia de algoritmo entonces puede ser explicada a través de un modelo de Markov.

El abordaje de Markov se aplica a la descripción del comportamiento aleatorio de un sistema que varía discretamente o continuamente con respecto al tiempo y espacio. Si son discretos se llaman cadenas de Markov. Si son continuos se llaman procesos de Markov. Para un proceso básico de Markov ser aplicable el comportamiento del sistema debe ser caracterizado por:

1. La falta de memoria. Esto significa que los futuros estados del sistema son independientes de todos los estados pasados excepto por el estado inmediatamente anterior. Sin embargo un comportamiento aleatorio futuro del sistema solamente depende de donde este en el presente y no donde estaba en el pasado y ni como el llegó en su presente posición.

2. El proceso debe ser estacionario u homogéneo. En un proceso estacionario la probabilidad de que haga una transición de un dado estado al otro es la misma por todos los tiempos pasados o futuros.

Entonces considerando esta pequeña introducción para que un algoritmo genético pueda ser analizado a través de una cadena de Markov él debe representar un proceso estacionario, sin memoria y que todos los estados puedan ser identificados o sea finito. Según Rudolph [138] [15] “la secuencia de mejores individuos encontrados en el tiempo por un algoritmo genético simple es una cadena de Markov irreducible en el espacio de

estados $\{0, 1, 2, \dots, n\}$ la cual no converge aunque la esperanza lo hace" [14] [15]. Pero esta situación cambia cuando se incorpora al algoritmo genético un mecanismo de elitismo.

El elitismo representa la capacidad del AG en identificar y guardar el mejor individuo j hallado en una determinada generación n que es insertado en la próxima generación hasta que sea remplazado por un nuevo individuo k mejor adaptado. Tal característica asegura la convergencia del algoritmo porque ahora no se corre el riesgo de perder la mejor solución una vez que sea encontrada.

Como visto en las gráficas anteriores un algoritmo genético sigue oscilando o explorando el espacio de búsqueda infinitamente hasta encontrar algún criterio que le ordene a parar. Por esto el algoritmo genético simple (no elitista) va encontrar o visitar el óptimo y lo perder infinitamente porque los operadores de cruce y mutación lo estarán impulsando continuamente a otro punto de la vecindad dentro del espacio de búsqueda. Tradicionalmente las probabilidades de cruce y mutación son muy pequeñas para evitar oscilaciones muy largas que tornarían el proceso de búsqueda más oneroso tanto del punto de vista computacional como del proceso de convergencia del algoritmo.

Las cadenas de Markov se caracterizan mediante un vector π y una matriz de transición P . El vector π representa la probabilidad de que la cadena se encuentre inicialmente en un estado determinado. Cada elemento (i, j) de P representa la probabilidad de pasar del estado "i" al estado j en una única transición. πP es un vector fila en el que cada componente es la probabilidad de estar en cada estado después de una transición. πP^k es un vector fila en el que cada componente es la probabilidad de estar en cada estado después de k transiciones. Dos estados "i" y "j" se comunican si al menos hay un camino de i a j y viceversa. Se dice que un estado es absorbente si no se comunica con ningún otro estado. Se dice que una cadena de Markov es irreducible si todos los estados se comunican entre sí. La ilustración 11 en la secuela presenta una representación gráfica de un sistema de tres estados. La transición del estado "i" (círculo superior de la derecha) para el estado "j" (círculo inferior de la izquierda) pasando por el estado "int" (círculo en el centro). Las setas representan transiciones tanto ascendientes como descendientes y a cada transición se asocia una probabilidad. Percibiese también que existe una probabilidad de no transición de uno estado para el otro o de permanencia en el estado actual representada por la seta pequeña.

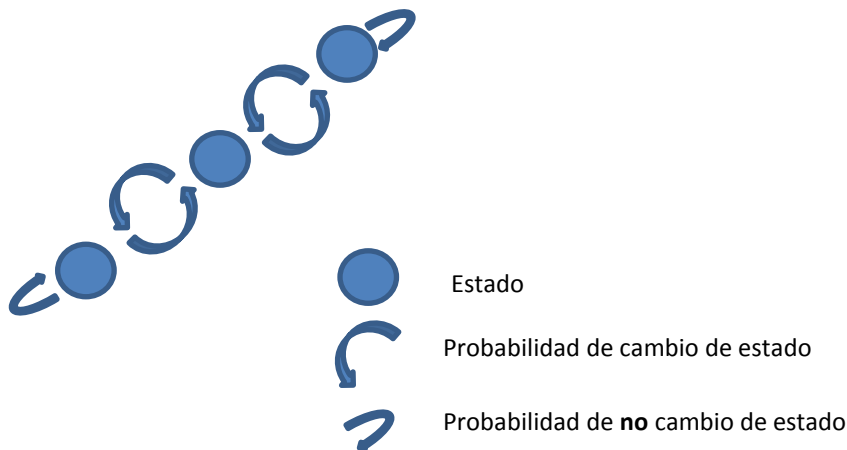


Figura 5 Diagrama de transición de estados Markov

Así para el análisis de los Algoritmos genéticos mediante una cadena de Markov se considera que un estado representa una determinada población. Y el conjunto de todos los estados es el conjunto de todas las posibles poblaciones [15]. Es claro que para casos realistas la cantidad de estados posibles es muy grande. A pesar de tener una dimensión muy alta, el espacio de estados es finito, por lo cual cumple que la cadena de Markov es finita. [94] Se puede representar a la matriz de transición como un grafo, donde un vértice corresponde a un estado y un enlace orientado del vértice i al vértice j una probabilidad de transición P_{ij} de i a j cuando ésta sea diferente de cero. En el caso estudiando cada individuo representa un vector de 95 variables de las cuales 45 son binarias (0,1) y 50 reales (float). Considerando 32 bits para cada una de las variables reales y un bit para las binarias la longitud " l " de un individuo será:

$$l = 45 \times 1 \text{ bit} + 50 \times 32 \text{ bits} = 1645 \text{ bits/individuo}$$

Así en conjunto de todas las posibles poblaciones de tamaño n corresponde al conjunto de todos los estados de la matriz de transición. Y nuestra matriz de transición tendrá la siguiente dimensión:

$$N = \frac{n + 2^l - 1}{2^l - 1}$$

Donde

n - tamaño de la población

l – longitud del individuo

Las columnas de la matriz de transición en la ilustración abajo corresponderán a una determinada población x . El valor de un elemento p_{ij} de la matriz corresponderá a la frecuencia de surgimiento de una combinación z de bits de un individuo. Cada línea corresponderá entonces a un vector que representa el número de apariciones de una determinada combinación de bits z en todas las poblaciones. Y la probabilidad de cambio de estado o surgimiento de la cadena z en la próxima población está representada pelo efecto conjunto de los operadores de mutación, cruce y selección.

		Población 0 hasta la población N			
		0	1	2.....	N
1 2..... 2^{n-1}			$P_{1,1}$...	$P_{1,N}$
			\vdots	\ddots	\vdots
		$P_{2^{n-1}-1,1}$...	$P_{i,j}$	

P= combinación de bits z 0 hasta la n

Ilustración 12 Matriz de transición

Un algoritmo genético que solo contenga selección y cruce no es irreducible ya que tiene estados absorbentes. Un estado absorbente para el AG representa que todos sus miembros o de una población son exactamente iguales. Los estados no absorbentes son considerados transitorios. Se ha demostrado que para un número suficientemente grande de generaciones, la cadena llega con probabilidad 1 a un estado absorbente (después de n pasos). [121-125]

Pero, existe una probabilidad no nula de que el estado absorbente al que se converja no sea óptimo(o no contenga al individuo óptimo). Si utilizamos un operador de mutación, los estados considerados como absorbentes se transforman en transitorios. Si la probabilidad de mutación es muy baja predomina el efecto del operador de cruzamiento sobre el de mutación. Bajo estas hipótesis el conjunto de estados generados será muy parecido al conjunto de estados originalmente planteados. Entonces, el operador de mutación es importantísimo, pues servirá para evitar la pérdida de información que puede impedir que se alcance la solución óptima. Incorporando el elitismo, que conserva la mejor solución encontrada a través de las generaciones, es posible demostrar que los AG elitistas convergen con probabilidad 1 a la solución óptima sin llevar en consideración su población inicial [14-15]. Y adicionalmente si el método de selección es aplicado tanto a

padres como a hijos el AG convergerá al óptimo global sin considerar su población inicial. Entonces según Kuri [15]:

1. Los AGs son una estrategia cuasi óptima para atacar problemas dinámicos.
2. Los AGs elitistas garantizan convergencia global en problemas estáticos.

Las consideraciones anteriores son válidas para los algoritmos genéticos multiobjetivos de acuerdo con Agapie [138]. Pero en este caso la convergencia estocástica al conjunto finito de elementos óptimos debe llevar en consideración una determinada métrica que mida la distancia entre puntos dentro de este mismo conjunto, que en nuestro caso es representada pelo operador “*crowding*”. Así de acuerdo con Agapie [138]

“Considerando A_t como la población de algún algoritmo evolutivo a cada iteración $t \geq 0$ y $F_t = f(A_t)$ asociado al conjunto imagen. Se dice que el algoritmo evolutivo converge con probabilidad 1 para todo el conjunto de elementos mínimos sí $d(F_t, F^) \rightarrow 0$ con probabilidad 1 con $t \rightarrow \infty$ de hecho converge con probabilidad 1 al conjunto de elementos mínimos sí $\square_{F^*}(F_t) \rightarrow 0$ con probabilidad 1 con $t \rightarrow \infty$*

Aquí F^ denota el conjunto de elementos mínimos”*

Está claro que $d(F_t, F^*) \rightarrow 0$ implica $\square_{F^*}(F_t) \rightarrow 0$. Mas en el primero caso el tamaño de la población que eventualmente crecerá por lo menos al tamaño del conjunto F^* . Sí (F, \preceq) es una cadena entonces $|F^*| = 1$ (para un único objetivo).

Más sí (F, \preceq) no es una cadena entonces (múltiples objetivos) entonces $|F^*|$ puede ser casi tan grande como el espacio de búsqueda, y como consecuencia, el tamaño de la población debe ser manejable y con un tamaño máximo. Desafortunadamente, este requerimiento hace con que los procedimientos de selección sean más complejos. Las demostraciones se encuentran en Agapie [139].

Como conclusión, la convergencia de un algoritmo genético elitista está asegurada aunque que el espacio de soluciones sea extremadamente grande (p.ej. $n \cdot 2^{1645}$ donde n es el número total de individuos) pero finito. Y la velocidad de convergencia está condicionada a un determinado número de generaciones y de individuos, considerando valores pequeños para las probabilidades de cruce y mutación [138], [139].

Así cuanto más grande sea este número más se acerca al “*optimum*” pero como consecuencia habrá más tiempo de cómputo. En las aplicaciones de ingeniería donde el “*optimum optimorum*” no es una condición *sine qua non* se puede aceptar buenas aproximaciones.

Al mantenerse constante el número de generaciones con tiempos de simulación diferentes observase una oscilación más grande de los valores de los objetivos en los experimentos de 1440 y 2160 horas en relación al experimento de 720 horas, graficas 5 y 6 respectivamente. Esto ocurrió debido a la presencia de más eventos estocásticos indeseados, como por ejemplo fallas, que provocaron una alteración en la disponibilidad promedio del sistema entre las simulaciones. Esta oscilación era esperada, considerando que algunos modos de fallas fueron definidos a través de modelos exponenciales con promedios por encima de 1000 horas (ver tabla 4). Sin embargo, en el caso investigado, para percibir la presencia de más eventos aleatorios, el sistema debe ser sometido a un tiempo más grande de simulación.

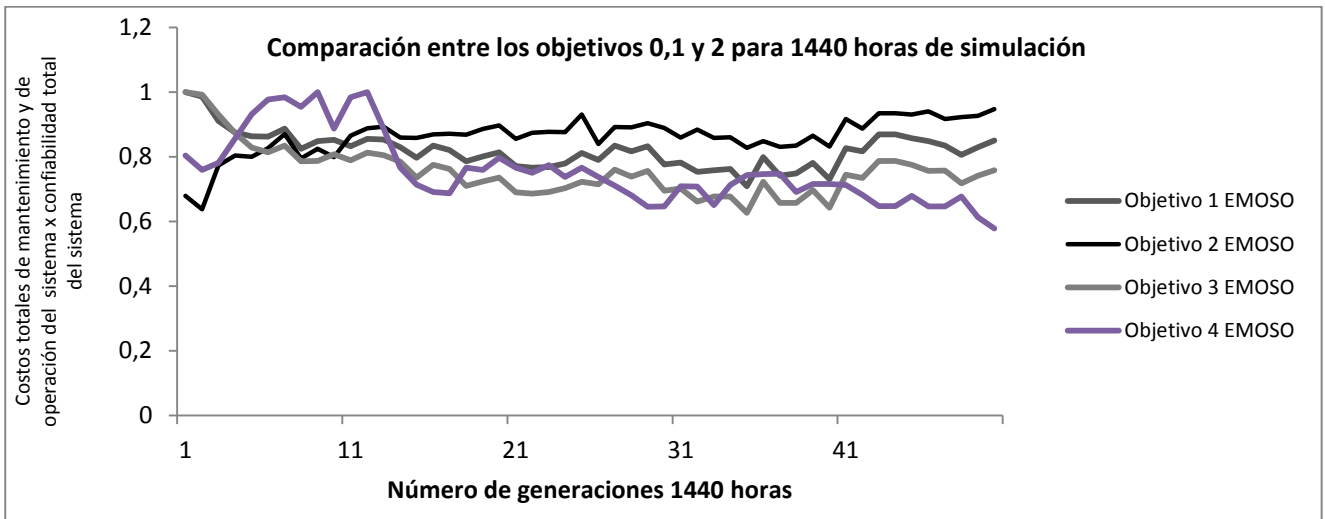
El tiempo de cómputo para todos los experimentos fue:

- 720 horas/50 generaciones – 14 horas 26 min.
- 1440 horas/50 generaciones – 21 horas 46 min.
- 2160 horas/50 generaciones - 23 horas20 min.
- 720 horas /100 generaciones - 26 horas 58 min
- 720 horas /150 generaciones - 51 horas 44 min

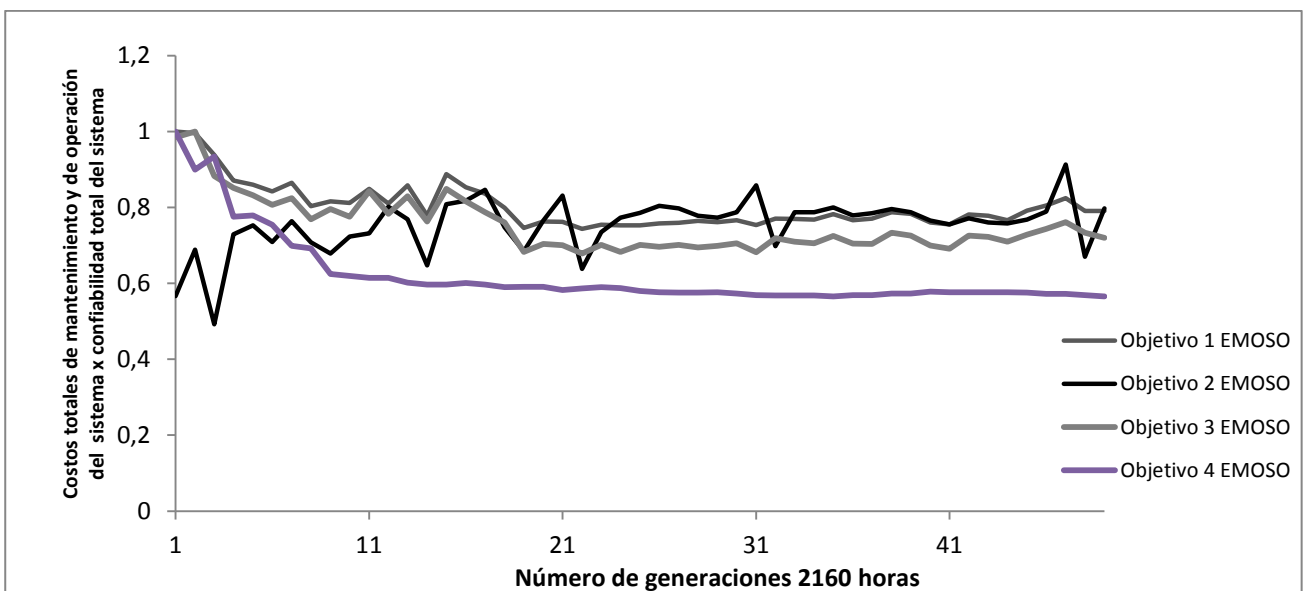
Estos resultados enseñan que, por lo tanto, la importancia del dimensionamiento adecuado de los tiempos de simulación de los sistemas sujetos a eventos estocásticos. Tiempos de simulación relativamente grandes propician una posibilidad más grande de ocurrencia de fallas que vengán interferir en el desempeño global del sistema. Ya tiempos relativamente cortos pueden no capturar esas mismas aleatoriedades.

Los resultados enseñan también la importancia del número de generaciones del experimento. Como esperado para procesos estocásticos, cuanto más grande es el número de simulaciones más grande es la posibilidad del AGE hallar soluciones más próximas del óptimo global.

Además del anteriormente dicho, los tiempos de cómputo de los experimentos fueron muy largos sería importante probarlos con una computadora con mejores características de “*hardware*”.



Gráfica 5 Comparación entre los objetivos 1, 2,3 y 4 para 1440 horas simuladas



Gráfica 6 Comparación entre los objetivos 1, 2,3 y 4 para 2160 horas simuladas

Pero el algoritmo genético sigue siempre mejorando las soluciones a medida que se le da más tiempo. Implementaciones más contemporáneas consideran, de acuerdo con [142], otros criterios de parada del AGE que podrían ser probados para alcanzarse un proceso de convergencia con tiempo de cómputo más pequeño. Estos criterios de parada son:

1. La variancia del desempeño "fitness" – En esta opción el AG se detiene una vez que la variancia del "fitness" de la población este debajo de un valor pre-determinado.
2. El mejor desempeño "fitness" – Esta opción el AG se detiene cuando el "fitness" del mejor individuo en una población rebasa un valor pre-determinado.

3. El desempeño "fitness" promedio- En esta opción el AG se detiene cuando el fitness promedio de la población rebasa un valor pre-determinado.
4. Cambio en el mejor desempeño "fitness" – En esta opción el AG se detiene cuando un cambio en el desempeño "fitness" de una población comparado a de una generación anterior este debajo de un valor especificado.
5. Cambio en el desempeño "fitness" promedio – En esta opción el AG se detiene cuando un cambio en el desempeño promedio "fitness" de una población comparado a de una generación anterior este debajo de un valor especificado.
6. Cambio en la variancia del desempeño "fitness" – En esta opción el AG se detiene cuando un cambio en la variancia del desempeño "fitness" de una población comparado a de una generación anterior este debajo de un valor especificado.
7. Cambio en el mejor objetivo – En esta opción el AG se detiene cuando el cambio en el valor del objetivo del mejor individuo de la población, en relación a generación anterior, este debajo de un valor especificado.
8. Cambio en el objetivo promedio - En esta opción el AG se detiene cuando el cambio en el valor del individuo promedio de la población, en relación a la generación anterior, este abajo del valor especificado.
9. Criterios relacionados con el número de fronteras, número de individuos en la primera frontera y cambios en las fronteras estos últimos para algoritmo basados en los conceptos de dominancia y fronteras de Pareto

6.3 Análisis de los resultados de los experimentos – multiobjetivo y mono objetivo

6.3.1 Consideraciones acerca del modelo mono objetivo

Un modelo simplificado a pesar de ser resuelto más fácilmente, no significa que sea menos complejo. No obstante, se el desarrollo de este modelo implicar en muchas simplificaciones comprometiese la acuidad de sus respuestas y, entonces no servirá al propósito.

La consistencia del modelo simplificado presentado, ítem 3.5, está basada en la apropiación de las mismas formulaciones de los costos que fueran aplicadas en el modelo de simulación, pero, sin la influencia de toda coyuntura estocástica cuya representación simulacional buscase emular.

Los costos del proceso en cuestión son originados a partir de una determinada configuración de máquinas que por su vez haz parte de un escenario de operación definido anteriormente.

Esto está basado en la formulación clásica de los modelos de optimización de la confiabilidad considerando el proceso de asignación de redundancias/confiabilidades que puede ser así definido [5-6]:

$$R_s = f(x, r)$$

Sujeto a

$$g_{ij} x_j, r_j \leq b_i, \quad \text{for } i = 1, \dots, m,$$

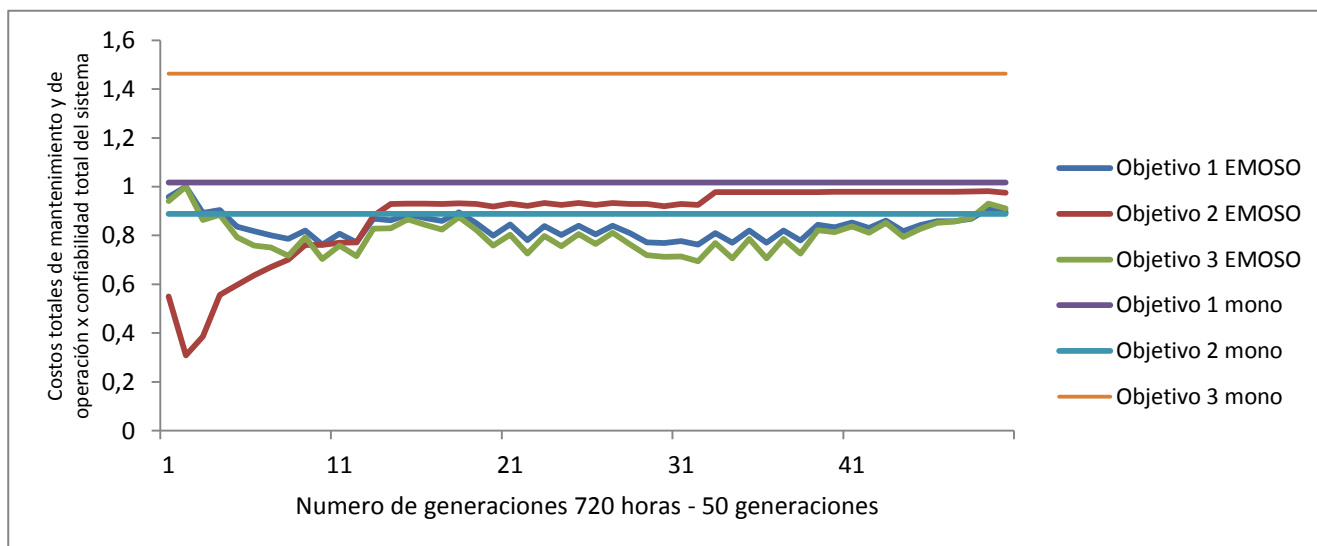
$$l_j \leq x_j \leq u_j, \quad \text{for } j = 1, \dots, n,$$

$$r_j^l \leq r_j \leq r_j^u, \quad \text{for } j = 1, \dots, n,$$

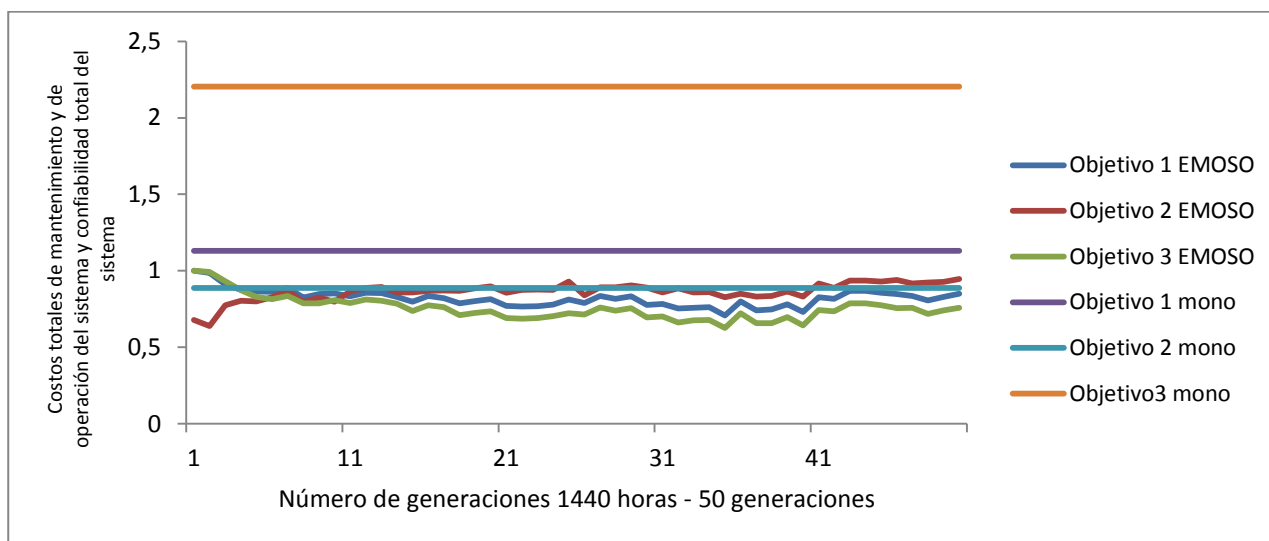
Vamos a denotar r_j y x_j como la confiabilidad del componente y el número de redundancias en la etapa j , respectivamente. Sea la función $f(x_1, \dots, x_n, r_1, \dots, r_n)$ que representa la confiabilidad del sistema correspondiente. Supóngase que hay m recursos y la cantidad disponible del recurso i es b_i para $i = 1, \dots, m$. Y que $g_{ij}(x_j, r_j)$ denotan el consumo de recurso i en la fase j para el nivel de confiabilidad del componente y nivel de redundancia r_j y x_j respectivamente.

En otras palabras, como se puede verificar en la formulación clásica de Kuo [5-6], la asignación de redundancias/confiabilidades determina un acrecimiento en el consumo de recursos tanto humanos, estructurales como de costos denotados por $g_{ij}(x_j, r_j)$ limitados por una cantidad b_i .

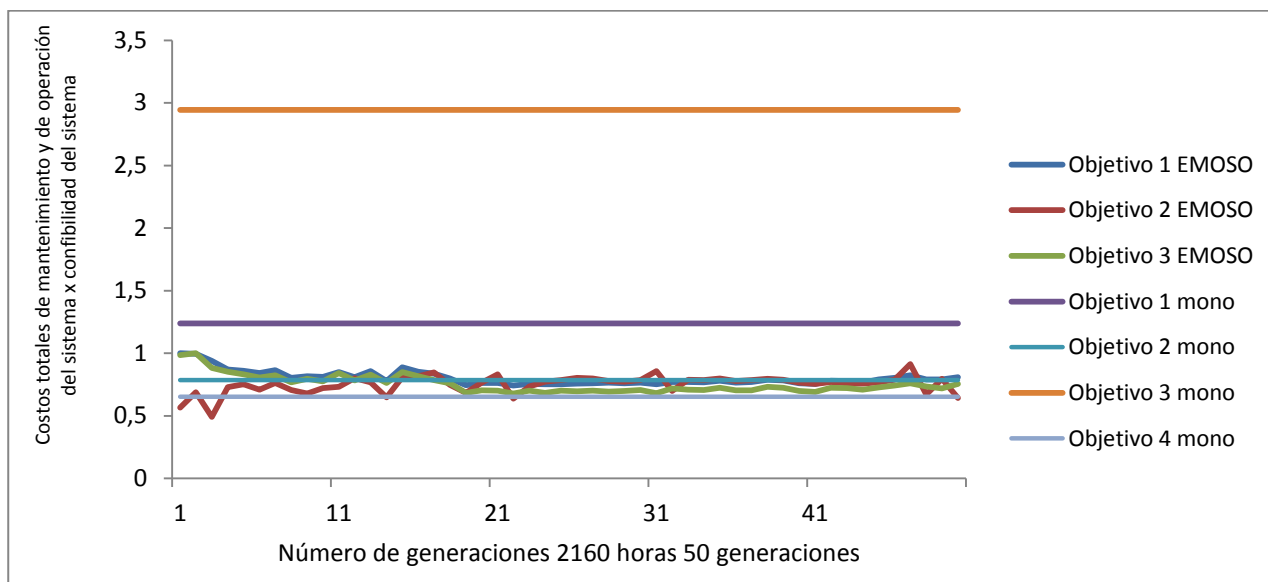
Las gráficas en la secuela, 7-9, enseñan la comparación de los resultados obtenidos en la estructura simulación/optimización – SIMOPT – y el modelo simplificado mono objetivo (MSMO). De un modo general la estructura SIMOPT superó el modelo MSMO en todos los objetivos.



Gráfica 7 Comparación entre los experimentos mono y multiobjetivos - 720 horas - 50 generaciones



Gráfica 8 Comparación entre los experimentos mono y multiobjetivos - 1440 horas - 50 generaciones



Gráfica 9 Comparación entre los experimentos mono y multiobjetivo - 2160 horas - 50 generaciones

En la tabla 6, fue hecha una comparación del objetivo 2 (Confiabilidad global del sistema). La columna Promedio representa el promedio de todas las generaciones de las corridas de los experimentos hechos en la SIMOPT.

Tabla 6 Comparación entre los experimentos (1,2 y 3) mono y multiobjetivo - Objetivo 2 - Confiabilidad global del sistema

	SIMOPT		MSMO	
		Promedio*	720/1440 horas	2160 horas
Objetivo 2	720/50	0.871855091	0.88	0.78
	1440/50	0.863537541	0.88	0.78
	2160/50	0.75215673	0.78	0.78
	720/100	0.853693481	0.88	0.78
	720/150	0.910970067	0.88	0.78

Dado la imposibilidad del modelo MSMO representar las características estocásticas inherentes del modelo idealizado en el trabajo él fue superado por la estructura SIMOPT, por lo tanto, a pesar de suministrar resultados de confiabilidad próximos de los promedios de las de las generaciones de individuos hallados en las corridas de la SIMOPT, esta comparación es inmaterial por dos motivos: uno que los promedios de la poblaciones son obtenidos de un conjunto de datos cuyos sus mejores individuos aparecen a partir de un determinado tiempo de ejecución del proceso, en otras palabras, estos promedios son obtenidos de un proceso que demora a estabilizar; otro

motivo es que los valores promedios de la SIMOPT no indican las mejores soluciones del proceso como se puede ver en las tablas 7-11.

6.3.2 – Consideraciones sobre la estructura SIMOPT

Fueran realizadas 8000 simulaciones. De todos los 8000 individuos generados, el AGE seleccionó 40 escenarios de operación distintos, que satisfacían todos los objetivos y restricciones del problema, tablas 7-11. Estos escenarios seleccionados fueran inicialmente agrupados según el número de máquinas asignadas en el proceso. De acuerdo con este criterio, observase que el número mínimo de asignaciones fue 15 y el máximo de 42, para un universo variando de 9 a 45 máquinas. Aisladamente, sin embargo, este criterio no permite una elección satisfactoria del escenario de operación.

Otro criterio que puede ser considerado, entonces, es la confiabilidad del escenario. La confiabilidad de los escenarios seleccionados presentase levemente oscilante con una tendencia de mejora con el aumento del número de máquinas asignadas (aumento de la redundancia). Es importante resaltar, sin embargo, que todos los escenarios seleccionados presentaron confiabilidades elevadas (arriba de 0.98) que de un punto de vista práctico hace este criterio también insuficiente para la toma de decisiones.

Tabla 7 Mejores escenarios del experimento 4 720 horas 50 generaciones

Experimento 4 (Corrida 4)		720 horas -50 generaciones				
		Objetivo 1 – Costo total de mantenimiento del sistema Objetivo 2 – Confiabilidad total del sistema Objetivo 3 – Costo de operación total del sistema Objetivo 4 – Número total de mantenedores del sistema				
Escenario de operación	Objetivo 1	Objetivo 2	Objetivo 3	Objetivo 4	Asignación de maquinas	Escenario de operación
1	2820206	0.998928	3624749	16	35	6.45
2	2792698	0.998885	3709403	16	35	5.92
3	1959979	0.997946	2079463	16	35	8.23
4	1606541	0.985626	1622457	5	31	9.76
5	1733869	0.987398	1676912	16	31	10.39
6	1809541	0.996716	1770991	16	32	11.19
7	1786686	0.996919	1775866	16	33	10.11
8	2468443	0.998576	3081645	16	34	6.69
9	2585959	0.998631	3326377	16	35	6.28
Promedios	2167227.2	0.995818	2481908	12	33.5	8.33

Tabla 8 Mejores escenarios del experimento 5 - 1440 horas - 50 generaciones

Experimento 5 (Corrida 5)	1440 horas - 50 generaciones					
Escenario de operación	Objetivo 1	Objetivo 2	Objetivo 3	Objetivo 4	Asignación de máquinas	Costo/producto
1	2914198	0.9993065	3147688	6	27	5.81
2	2043527	0.9985866	2074511	7	27	6.81
3	2742798	0.9987716	2901673	6	27	6.13
4	1669170	0.9797631	1641688	6	20	7.11
5	1992993	0.9837161	2040098	15	21	6.30
6	2010620	0.9800117	1993446	7	20	7.19
7	2863146	0.9991417	3064383	6	29	5.83
8	2771984	0.9990900	2931642	6	28	6.18
9	2896160	0.9993100	3130280	6	29	5.77
10	2788534	0.9988900	2915434	6	27	7.10
Promedios	2469313	0.9936590	2584084	7.1	25.50	6,42

Tabla 9 Mejores escenarios del experimento 6 - 2160 horas - 50 generaciones

Experimento 3 (Corrida 3)	2160 horas - 50 generaciones					
Escenario de operación	Objetivo 1	Objetivo 2	Objetivo 3	Objetivo 4	Asignación de maquinas	Escenario de operación
1	1658352	0.981308	1623417	6	23	10.11
2	1923583	0.985386	1899560	6	20	11.15
3	2157052	0.993085	2287982	6	24	7.84
4	1965154	0.989701	2058740	6	21	7.91
5	2217531	0.994180	2399669	6	25	7.24
6	2405210	0.995831	2726515	6	25	6.10
7	2145893	0.993032	2287960	6	24	7.56
8	2031733	0.991144	2119967	6	22	8.29
9	2239416	0.994185	2418995	6	25	7.30
10	2404594	0.995893	2728048	6	25	6.10
Promedios	2114856	0.991330	2255058	6	23.4	7.96

Tabla 10 - Mejores escenarios experimento 7 720 horas - 100 generaciones

Experimento 7 (Corrida 7)	720 horas – 100 generaciones					
Escenario de operación	Objetivo 1	Objetivo 2	Objetivo 3	Objetivo 4	Asignación de	Escenario de operación
1	1967423	0.999746	2162364	6	36	6.23
2	1641490	0.999063	1801222	5	35	5.65
3	1744774	0.999701	1955186	6	36	5.40
4	1696361	0.999691	1893462	6	36	5.40
5	1691796	0.999581	1885273	6	36	5.42
6	2014479	0.999761	2166809	6	35	7.00
Promedios	1792721	0.999591	1977386	6	35.6	5,85

Tabla 11 Mejores escenarios experimento 8 720 horas 150 generaciones

Experimento 8 (Corrida 8)	720 horas – 150 generaciones					
Escenario de operación	Objetivo 1	Objetivo 2	Objetivo 3	Objetivo 4	Asignación de	Escenario de operación
1	1972618	0.993765	1986444	6	31	7,53
2	1929498	0.998884	2103543	6	34	8,31
3	1919545	0.997096	1991977	6	32	8.18
4	1729870	0.984319	1814671	15	30	9,38
5	2274398	0.999943	2396470	6	42	9,09
Promedios	1965186	0.994803	2042621	7.8	33.8	8.50

Delante del expuesto, el tomador de decisión puede considerar otras variables de interés como el costo/producto, costo de mantenimiento (Objetivo 1) y los costos de operación (Objetivo 3). Con este nuevo criterio observase que el costo mínimo encontrado fue de 5.4 (U.M.) (experimento 7 720/100/3 y 720/100/4) tabla 10 y el máximo de 11.19 U.M. (experimento 1 720/50/6) tabla 7, correspondiendo a 36 y 32 máquinas asignadas respectivamente.⁴ Para la asignación de 36 máquinas el número de mantenedores encontrado fue 6 contra 16 de la asignación de 32 máquinas, por ejemplo.

Y, agregando a esto criterio que el número de mantenedores más pequeño deseado sea inferior a 7 con costo/producto inferiores a 6.0 U.M. Con esto y las soluciones presentadas en las tablas 7-11. Y procurando responde la pregunta dejada en el ítem 1.2 llegase a conclusión que los escenarios más apropiados (óptimos) para el caso idealizado en cuestión serían los escenarios de operación de la tabla 12 en la secuela:

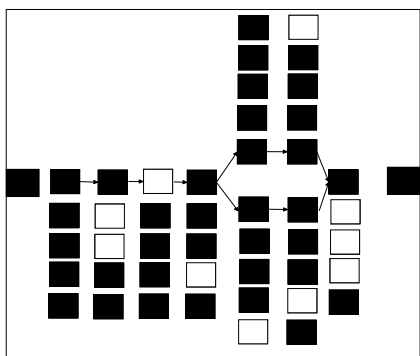
⁴ U.M. = unidades monetarias

Tabla 12 Mejores escenarios hallados después de la agregación de información

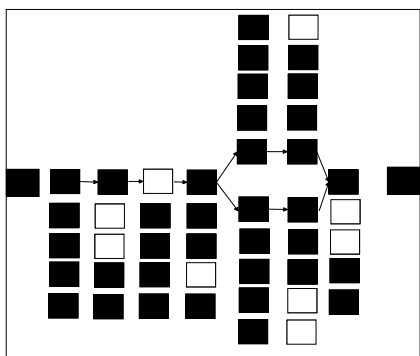
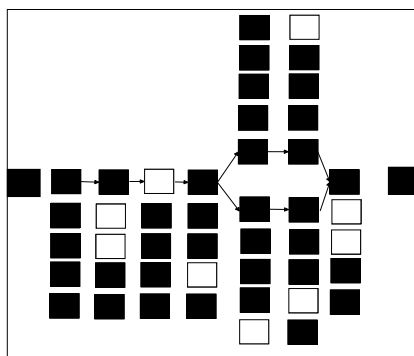
Tiempo de Simulación /generación	Número del escenario de operación	Objetivo 1	Objetivo 2	Objetivo 3	Objetivo 4	Asignación de máquinas	Costo/producto
720/100	2	1641490	0.999063	1801222	5	35	5.65
720/100	3	1744774	0.999701	1955186	6	36	5.40
720/100	4	1696361	0.999691	1893462	6	36	5.40
720/100	5	1691796	0.999581	1885273	6	36	5.42
1440/50	1	2914198	0.9993065	3147688	6	27	5.81
1440/50	7	2863146	0.999142	3064383	6	29	5.83
1440/50	9	2896160	0.999310	3130280	6	29	5.77

Cabe observar que, para números de asignaciones de máquinas muy próximos, ocurrió una diferencia entre sus costos. Esto evidencia una característica interesante del proceso que no solamente la cantidad de máquinas asignadas fue tomada en cuestión como también la posición de cada una de ellas en el sistema. Esto produce alteraciones en la producción y expone el sistema a la presencia de fallas que pueden causar, por ejemplo, un tiempo más grande de paralización de una operación. En estos casos, a pesar de cuantitativamente muy próximas, la combinación de máquinas es diferente y, consecuentemente, el desempeño también. Como ilustración fueron dibujadas las configuraciones de máquinas A, B, C, D, E y F correspondiente a los escenarios 2, 3, 4, 7 y 9 respectivamente de la tabla 12. Considerar los cuadrados negros como máquinas asignadas y los cuadrados blancos como máquinas no asignadas.

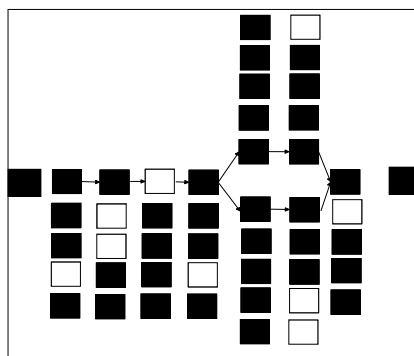
A Escenario de operación 2 – 720 horas 100 generaciones



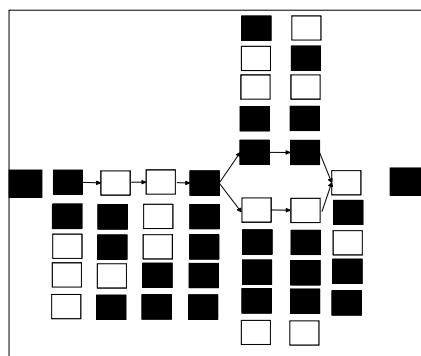
B – Escenario de operación 3 – 720 horas 100 generaciones



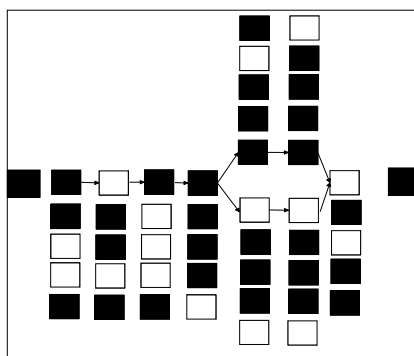
C –Escenario de operación 4 720 horas – 100 generaciones



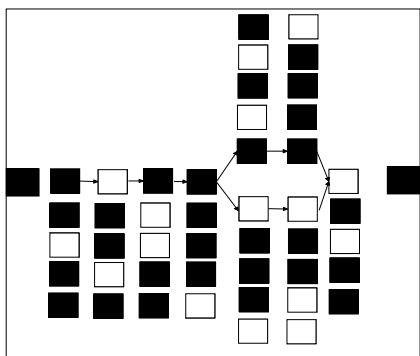
D – Escenario de operación 5 – 720 horas 100 generaciones



E – Escenario de operación 1 – 1440 horas 50 generaciones



F – Escenario de operación 7 – 1440 horas 50 generaciones



G – Escenario de operación 9 – 1440 horas 50 generaciones

Figura 6 Demostración grafica de la tabla 12

6.3.3 Especulaciones acerca de las funciones de ajuste estadístico:

Considerando los experimentos multiobjetivos realizados, como mencionado no capítulo 4, en la secuela:

- **Experimentos 4, 5 y 6** de 720, 1440 y 2160 horas simuladas con 50 generaciones de 20 individuos (1000 individuos generados por corrida) respectivamente
- **Experimento 7** de 720 horas simuladas con 100 generaciones de 20 individuos (2000 individuos generados)
- **Experimento 8** de 720 horas simuladas con 150 generaciones de 20 individuos (3000 individuos)

Cabe recordar que cada individuo corresponde a un escenario de operación y que cada conjunto de 720 horas corresponde a un mes de operación sin interrupción del sistema. Estos individuos fueran sometidos al proceso de evolución del AGE lo cual, de acuerdo con el ya discutido, es guiado por los operadores de reproducción.

Estos operadores son responsables por la amplitud y diversidad de la búsqueda y tuvieran sus probabilidades de ocurrencia mantenidas constantes para garantizar una cierta homogeneidad de búsqueda entre los experimentos multiobjetivos. De esta manera, se evitaran oscilaciones muy grandes dos resultados entre los 5 experimentos multiobjetivos. Los experimentos produjeran una grande cantidad de datos agrupados por generaciones.

Como discutido anteriormente el AGE sigue siempre mejorando las soluciones a medida que se le da más tiempo y su convergencia está condicionada a un determinado número de generaciones, de individuos por generación, a las probabilidades de los operadores de reproducción y método de selección empleado. Así la convergencia está asegurada aunque que el espacio de soluciones sea extremadamente grande pero finito.

El análisis estadístico empieza con la extracción de una tabla de resumen estadístico para cada experimento. Se observa en esta tabla las estadísticas descriptivas que representan: el desempeño promedio o "*fitness*" promedio; una medida del error típico del desempeño promedio; la mediana y variancia del desempeño; el error estándar, el desempeño máximo y mínimo; el rayo del intervalo de confianza para el desempeño promedio suponiendo población normal de varianza desconocida; y el coeficiente de asimetría y la medida de curtosis o apuntamiento. Esperase que estos últimos estén dentro del intervalo $[-2,2]$ para verificar si efectivamente la población es normal.

La tabla 13, presenta un ejemplo de la tabla de resumen estadístico del experimento 4 50 generaciones y 720 horas de simulación:

720 horas - 50 generaciones				
	Objetivo 1	Objetivo 2	Objetivo 3	Objetivo 4
Promedio	0,644696851	0,868702328	0,53914213	0,60324149
Error típico	0,004462557	0,00927578	0,00544317	0,00755577
Mediana	0,584664281	0,987294675	0,46536145	0,80132596
Moda	0,523552234	0	0,41033134	0,80301174
Desviación estándar	0,141118438	0,29332593	0,17212804	0,23893447
Varianza de la muestra	0,019914413	0,086040101	0,02962806	0,05708968
Curtosis	0,586779017	4,611334655	0,35816402	-1,6336841
Coefficiente de asimetría	0,735529362	-2,5094529	0,99968533	-0,4461652
Rango	0,907358567	1	0,99999985	0,74208176
Suma	644,696851	868,7023277	539,14213	603,241492
Cuenta	1000	1000	1000	1000
Mayor(1)	1	1	1	1
Menor(1)	0,092641433	0	1,468E-07	0,25791824
Nivel de confianza (95,0%)	0,00875706	0,018202248	0,01068135	0,014827
Coefficiente de Pearson	0,425405574	-0,4043023	0,42863836	-0,8290326

Tabla 13 - Tabla de resumen estadístico del experimento 4

Verificase en la tabla 13 que las distribuciones de los desempeños de todos los objetivos no son simétricas y por lo tanto invalida el uso de métodos paramétricos como t-estudiante, ANOVA, etc. asumiendo que x_1, x_2, \dots, x_n muestras representan una secuencia de valores independientes e idénticamente distribuidos [143]. Este suceso fue identificado en todos los experimentos de acuerdo con la tabla 14, utilizando el coeficiente de simetría de Pearson en la secuela:

Experimento	Objetivo 1	Objetivo 2	Objetivo 3	Objetivo 4
4 -720 horas 50 generaciones	0,42540557	-0,40430230	0,42863836	-0,82903261
5-1440 horas 50 generaciones	0,30704225	-0,21197779	0,27868393	0,72475183
6-2160 horas 50 generaciones	0,28537810	-0,32405147	0,13513543	0,07178890
7-720 horas 100 generaciones	0,52096536	-0,47444772	0,50979017	0,26978468
8-720 horas 150 generaciones	0,21856782	-0,32311888	0,26588966	-0,14976595

Tabla 14 Coeficiente de simetría de Pearson de todos los experimentos

Observase también en la tabla 14 que el grado de asimetría sigue el mismo estándar para todos los experimentos, o sea, las corridas con asimetría a derecha siguen con la misma tendencia, mismo variando el número de generaciones o tiempo de simulación.

Entonces dado que no se puede tratar los datos con métodos paramétricos se propone una combinación de un abordaje de análisis exploratoria de datos, en particular las gráficas de caja y extensión, con un simple teste no paramétrico como por ejemplo el Kruskal_Wallis [142].

Una gráfica de caja y extensión encierra el rango intercuartil de los datos en una caja que tiene la mediana dibujada dentro. El rango intercuartil de los datos tiene como extremos el percentil 75 (cuartil superior) y el percentil 25 (cuartil inferior). Las extensiones que indican observaciones extremas son indicadas por las líneas que están abajo o encima de la caja (no caso del rango de datos estar representado en el eje Y). Con esta grafica se puede identificar el despliegue de datos en relación al centro de localización (mediana), la variabilidad y el grado de asimetría. [142]

La prueba de Kruskal_Wallis es una generalización de la prueba de la suma de rangos para el caso de $k > 2$ muestras. Se utiliza para probar la hipótesis nula H_0 de que k muestras independientes son de poblaciones idénticas. Tratase de un procedimiento no paramétrico para probar la igualdad de promedios o medianas en el análisis de variancia de un factor cuando el experimentador desea desechar la suposición que las muestras se seleccionaron de poblaciones normales. [142]

Tratándose del proceso de convergencia del AGE fueran probados dos factores distintos de influencia en el proceso: uno influenciando las variables exógenas de los experimentos y otro las variables endógenas. El primero fue el tiempo de simulación diferente, factor exógeno, para experimentos con un mismo número de generaciones; el segundo fue el número de generaciones del AGE, factor endógeno, para experimentos con un mismo número de horas de simulación. En ambos se consideraron constantes, el número de individuos por generación y las probabilidades de ocurrencia de los operadores genéticos.

Evaluando el primero factor exógeno a través de las gráficas de caja y extensión fue evidenciada la influencia de él en la dispersión de las respuestas y por la característica acentuada de asimetría para los tiempos de simulación más grandes. Esto comprueba la aseveración anterior de que sometándose el sistema a un tiempo más

grande de simulación fue posible la percepción de factores estocásticos indeseables como por ejemplo las fallas. Las gráficas fueron generadas a partir de todos los desempeños de los objetivos generados en el experimento para 50 generaciones, 1,000 en el total. Por ejemplo, las gráficas en las figuras 7, 8 y 9 enseñan los resultados del desempeño del objetivo 2, maximización de la confiabilidad del sistema, para 50 generaciones, 720, 1440 y 2160 horas respectivamente en la secuela:

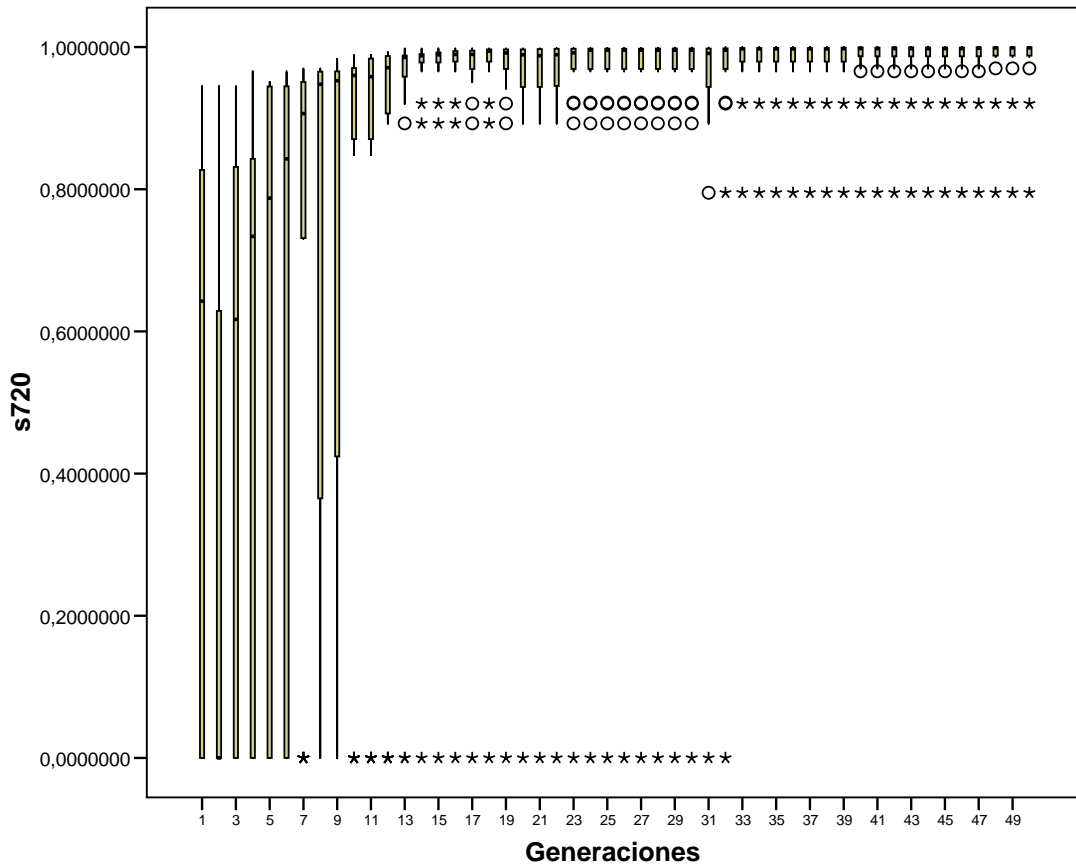


Figura 7 - Grafica de cajas y extensión - Objetivo 2 720 horas 50 generaciones

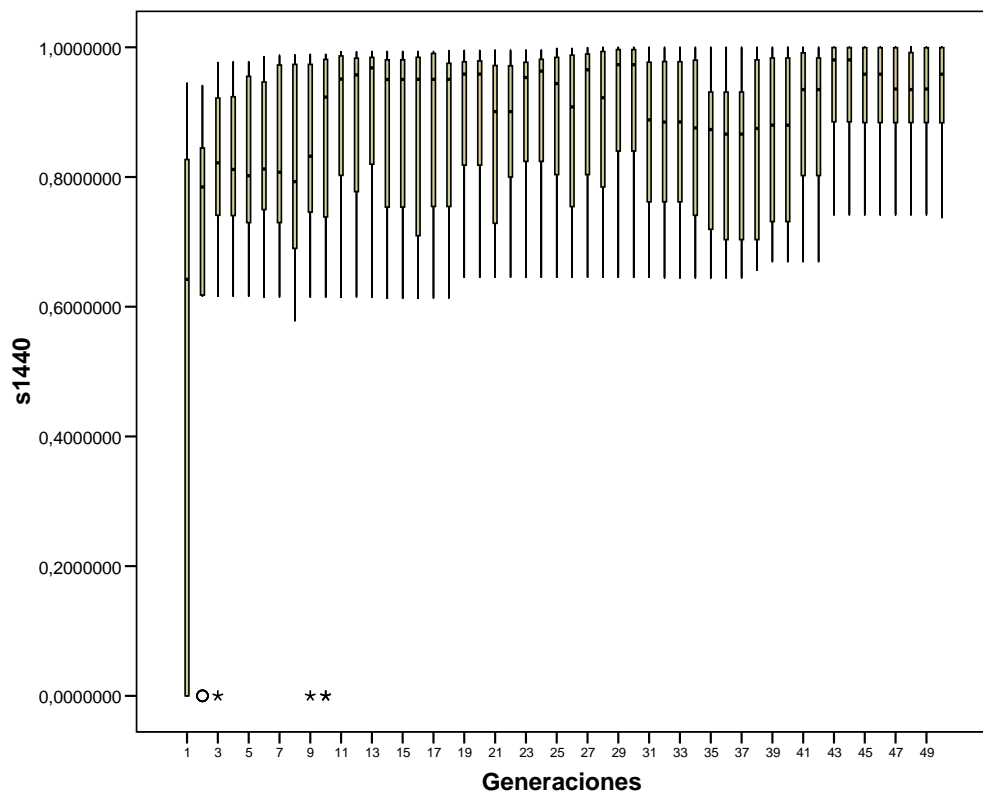


Figura 8 Grafica de caja y extensión para el Objetivo 2 - 1440 horas - 50 generaciones

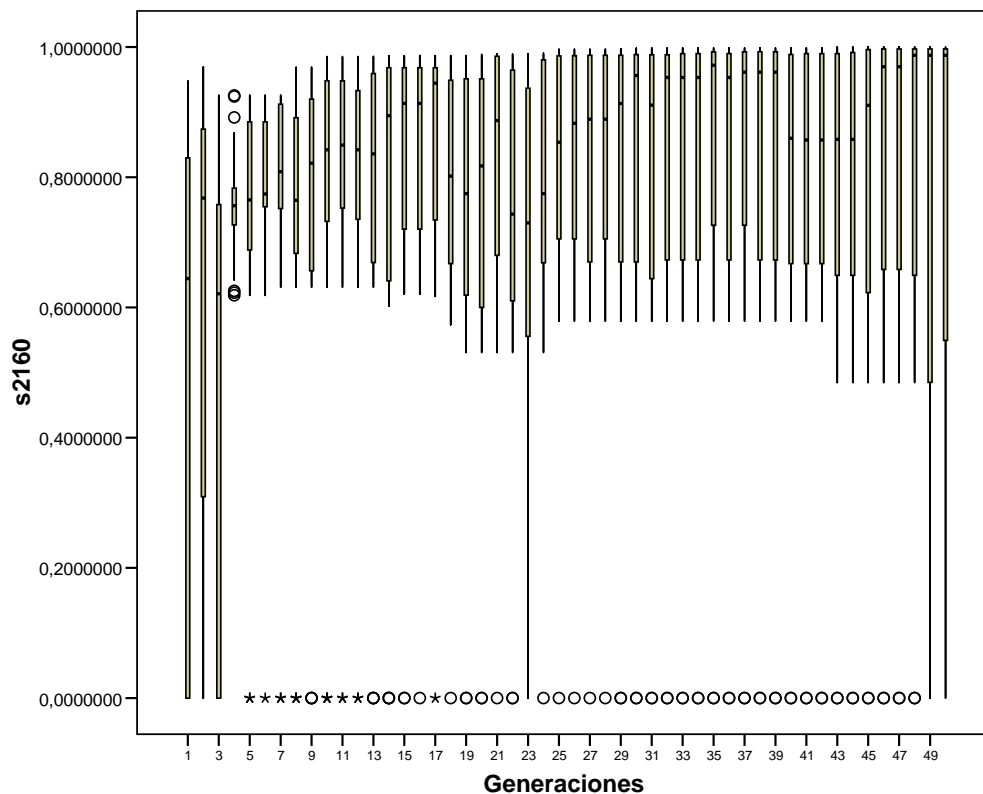


Figura 9 Grafica de caja y extensión para el Objetivo 2 - 2160 horas - 50 generaciones

Y para aplicar el teste de Kruskal-Wallis fueron hechas las consideraciones en la secuela:

1 Hipótesis

$H_0: M_1=M_2=M_3$ (Las medianas de las muestras son iguales)

H_a : Las medianas son diferentes

2. Nivel de significancia $\alpha= 0.05$

3. Región de no aceptación

Rechazar la hipótesis nula si $p\text{-value} \leq 0.05$

De acuerdo las tablas XXX y VVV en la secuela:

Test Estadísticas (a,b)

	s720	s1440	s2160
Chi-Square	444,359	136,199	88,293
df	49	49	49
Asymp. Sig.	,000	,000	,000

Tabla 15 Tabla de puntuación del teste de Kruskal-Wallis – proba del factor exógeno

a Kruskal Wallis Test

b Grouping Variable: Generaciones

Y como el $p\text{-value}=0.00 \leq 0.05$ la hipótesis nula es rechazada.

Se pudo afirmar que con el nivel de significancia $\alpha=0.05$ existe evidencias suficientes para concluir que existe una diferencia significativa entre los tres tiempos de simulación basado en la tabla 15 de puntuación.

Evaluando el segundo factor endógeno la influencia del número generaciones observase la tendencia de disminución de la dispersión del desempeño de los objetivos en los experimentos con número de generaciones más grande de acuerdo con las gráficas de caja y extensión en las figuras 10 – 12, para el desempeño del objetivo 1, minimización del costo total de mantenimiento del sistema, para 720 horas y 50, 100 y 150 generaciones respectivamente, en la secuela:

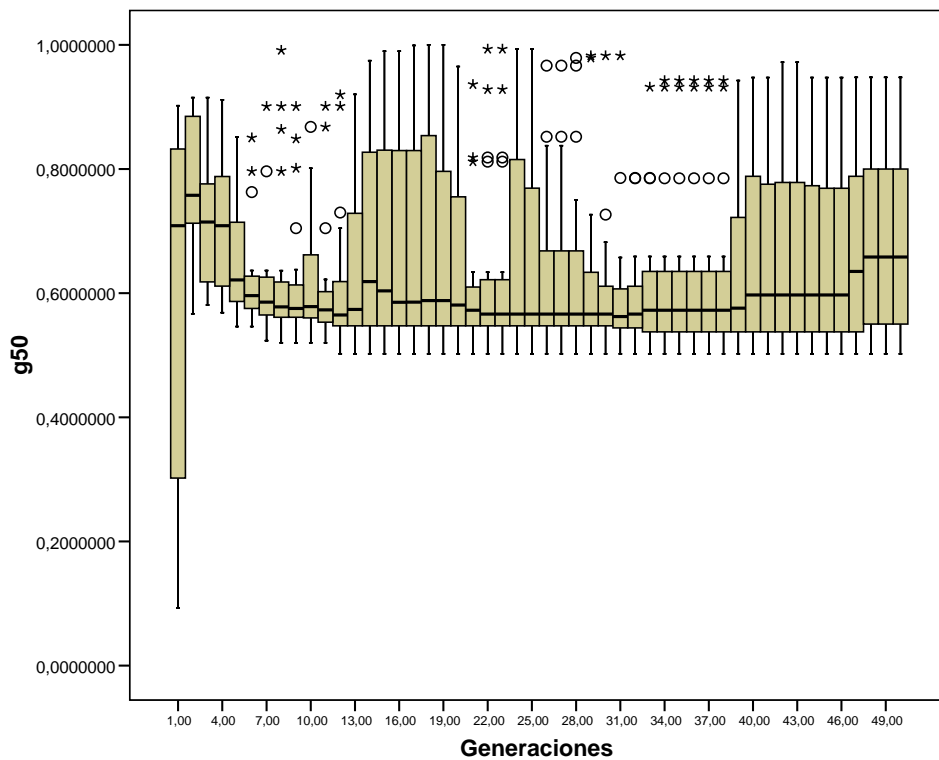


Figura 10 Grafica de caja y extensión para el objetivo 1 720 horas - 50 generaciones

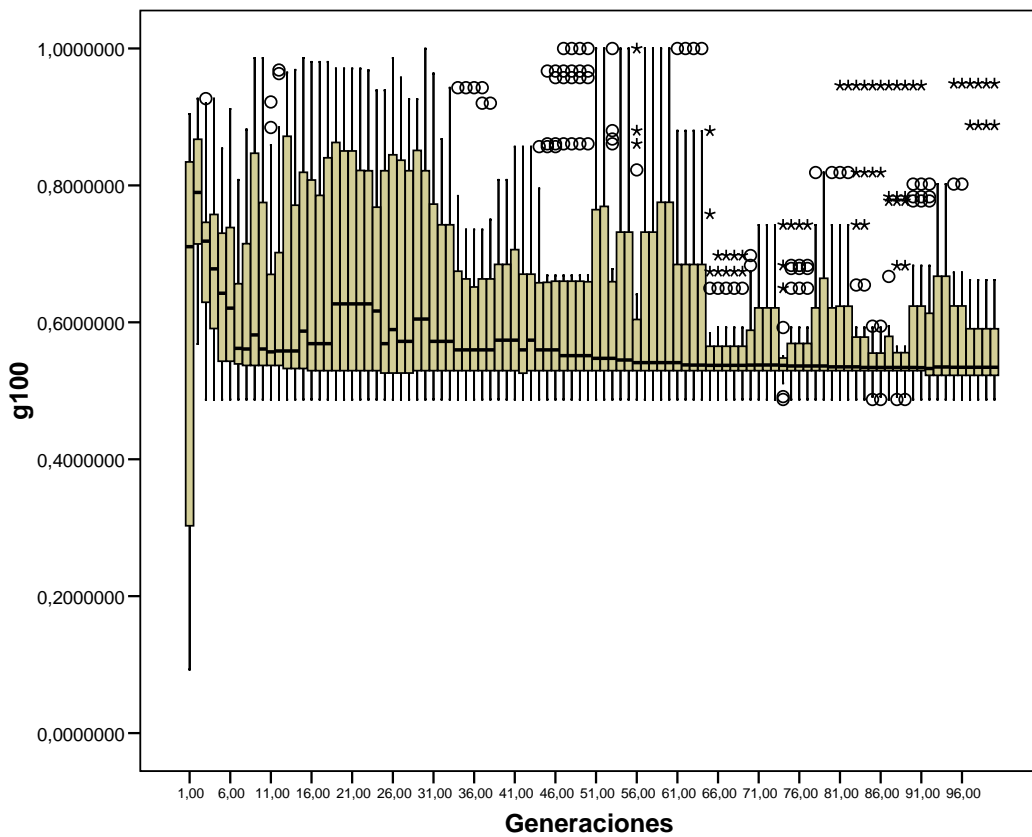


Figura 11 Grafica de caja y extensión para el objetivo 1 720 horas - 100 generaciones

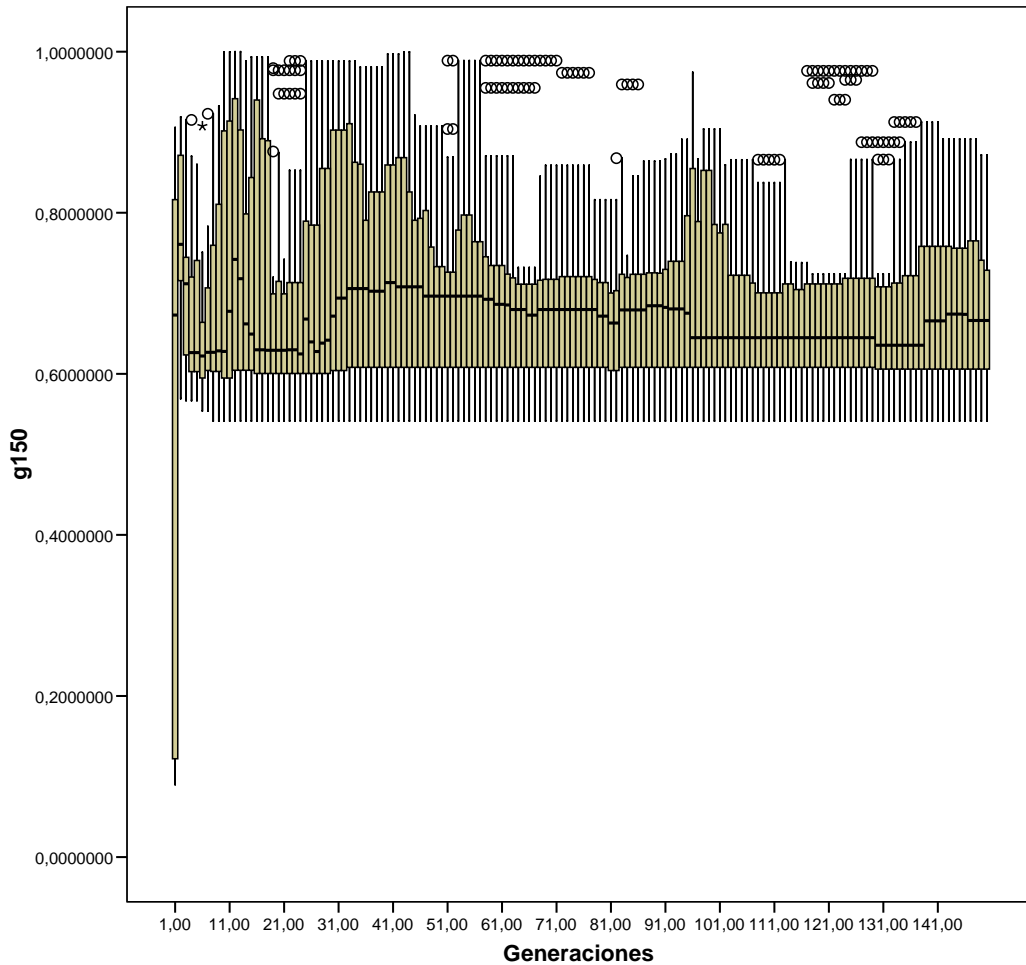


Figura 12 - Grafica de caja y extensión para el Objetivo 1 - 720 horas - 150 generaciones

Considerando la tabla de resumen estadístico para las tres poblaciones correspondiendo al desempeño del objetivo 1 para 50, 100 y 150 generaciones observase que además del grado de asimetría comprobado las variancias son diferentes y posiblemente indicando que se está en una condición de heterocedasticidad, pero con son más de dos experimentos el analice con las gráficas caja y extensión y la tabla de resumen estadístico complementados con el teste de Kruskal_Wallis sigue siendo empleado.

Así aplicando el teste de Kruskal-Wallis para verificar si hay influencia del número de generaciones en los experimentos se concluye que con el nivel de significancia $\alpha=0.05$ existe evidencias suficientes para concluir que existe una diferencia significativa entre los tres tamaños de generación basado en las tablas de resumen estadístico y de puntuación del teste de Kruskal-Wallis tabla 16 y 17 respectivamente. Y con esto el número de generaciones influencia el proceso de convergencia.

		Generaciones		
Estadísticas		g50	g100	g150
N	Estadística	1000	2000	3000
Rango	Estadística	.9073586	.9071227	.9099224
Mínimo	Estadística	.0926414	.0928773	.0900776
Máximo	Estadística	1.0000000	1.0000000	1.0000000
Suma	Estadística	644.6968509	1234.6971491	2066.1347429
Promedio	Estadística	.644696851	.617348575	.688711581
	Error estándar	.0044625568	.0030737265	.0022240204
Desviación estándar	Estadística	.1411184375	.1374612260	.1218146119
Variancia	Estadística	.020	.019	.015
asimetría	Estadística	.736	1.103	.665
	Error estándar	.077	.055	.045
Kurtosis	Estadística	.587	.678	.842
	Error estándar	.155	.109	.089

Tabla 16 - Tabla de resumen estadístico para el Objetivo 1 720 horas - 50, 100 y 150 generaciones

Teste estadístico (a,b)

	g50	g100	g150
Chi-Square	69,121	152,801	66,700
df	49	99	149
Asymp. Sig.	,031	,000	1,000

Tabla 17 - Tabla de puntuación para el teste de Kruskal-Wallis del factor endógeno

a Kruskal Wallis Test

b Grouping Variable: Generaciones

Y concluyendo existe influencia del número de generaciones en la convergencia del proceso de simulación/optimización.

Las gráficas, 2-6, de los desempeños promedios de los objetivos proporcionaron un análisis sencillo del proceso de convergencia del AGE. Y específicamente para el caso estudiado el modelo de optimización mono-objetivo proporcionó algunas cotas de comparación, mismo con sus debilidades. El análisis de convergencia fue complementada con un análisis no paramétrico concluyendo que el AGE siempre convergirá a un valor de desempeño mejor adaptado y factores como el tiempo de simulación y número de generaciones influyen tanto en la calidad de las soluciones como también en el tiempo de cómputo necesario.

Todos los experimentos, 4 -8, presentarán el mismo estándar y respondieron de la misma manera a la aplicación de los factores endógeno y exógeno. Pequeñas variaciones ocurrirán pero en general presentarán el mismo comportamiento estadístico.

Ciertamente ajustes estadísticos que llevan en cuenta la característica estadística muy particular de los datos generados por algoritmos evolutivos son necesarios. La razón para esto está en la forma peculiar como el proceso de selección de los individuos de una generación para otra es realizado resultando en datos que acaban sesgados, asimétricos y no normalmente distribuidos [143]. Concluyendo esto imposibilita la aplicación de métodos paramétricos de análisis estadístico. Lahoz-Beltra y Perales-Gravan [143] proponen tratar estas características de los experimentos con algoritmos evolutivos a través de un protocolo estadístico sintetizado en la tabla 18:

Tabla 18 Protocolo de análisis estadístico para experimentos de computación evolutiva

Evaluación del desempeño	Gráficas robustas de desempeño y tabla de resumen estadístico		
	$K_e=2$ experimentos	Gráficas múltiples de caja y extensión	$\sigma_i=\sigma_j$ <i>Prueba de Mann-Whitney (Wilcoxon)</i>
		Tabla de resumen estadístico	$\sigma_i\neq\sigma_j$ Prueba de Wilcoxon estudentizada
	$K_e>2$ experimentos	Gráficas múltiples de caja y extensión Tabla de resumen estadístico Prueba de Kruskal-Wallis Prueba de Dunn	

Fuente – Lahoz-Beltra y Perales-Gravan [127]

7. Conclusión

Este trabajo trató de un problema de asignación de redundancias y confiabilidades en un sistema de producción automatizado. El objetivo es mejorar la confiabilidad global del sistema asignando componentes alternativas (redundancias) asociadas en paralelo a cada componente original y, al mismo tiempo, minimizar el empleo de mantenedores y costos de mantenimiento/operación asociados.

El proceso de simulación/optimización utilizado hizo uso de un modelo construido en términos de variables y restricciones. No existe, como en los problemas de optimización convencionales, una función (o conjunto de funciones) representando el sistema. De esta manera, el modelo puede asumir el grado cualquiera de complejidad, pudiendo, sin embargo, representar fidedignamente situaciones reales industriales.

De esta manera, la metodología empleada está íntimamente ligada a las restricciones impuestas al modelo, las cuales son fruto del juicio del modelador. Así siendo, el proceso no solamente beneficiase, más de facto depende de la corrección (calidad) y del trabajo del modelador.

Del universo de simulaciones realizadas (8,000) fueron seleccionadas por el proceso genético solamente 40 escenarios de operación factibles. Así, el abordaje propuesto permite la generación de un conjunto reducido y exequible de soluciones (bastando alterar los límites de las restricciones), minimizando la cantidad de escenarios de operación y consecuentemente facilitando la toma de decisiones.

La metodología empleada fue eficaz en el sentido de encontrar soluciones factibles, a pesar de la existencia de múltiples objetivos en conflicto, restricciones, inter relaciones sistémicas y aleatoriedades. Delante de esto y considerando todavía el altísimo grado de confiabilidad de las soluciones atingidas (>0.98) y el grado de generalidad del modelo desarrollado, puede-se afirmar que la metodología propuesta tiene amplia posibilidad de aplicación en situaciones reales industriales.

A pesar de las respuestas encontradas sean satisfactorias delante de las restricciones empleadas, el proceso/modelo desarrollado permite análisis más profundizados. Esto está relacionado con las múltiples posibilidades que las variables más específicas, que pueden ser asignadas en el modelo, posibilitarían analizar.

Como desdoblamientos futuros de esto trabajo se puede considerar la utilización de sistemas de producción con estructura más compleja, además de la utilización de otros objetivos como:

- Optimización del inventario de partes para el mantenimiento;
- Optimización del número de inspecciones de mantenimiento con el objetivo de reducir los tiempos de paralización del sistema;
- Optimización de la calendarización de tareas(*job shop problem*) ; y
- Optimización de los parámetros de operación con el objetivo de atender a las alteraciones de los niveles de productividad.

8. Referencias principales:

- [1] TAHA, HAMDY A.-**Investigación de Operaciones**. Alfaomega, 5ed, 1995.
- [2] HILLIER, FREDERICK S. AND LIEBERMAN, GERALD J. – **Introducción a La Investigación de operaciones**. México. McGraw-Hill Interamericana Editores AS de CV 1997.
- [3] LEVITIN, GREGORY; LISNIANSKI ANATOLY. **Multi-state system reliability. Assessment, optimization and applications**. London,UK, World Scientific Publishing Co, 2003.
- [4] BILLINTON, ROY; ALLAN, RONALD N. **Reliability evaluation of engineering systems. Concepts and techniques**. Springer, 2nd edition, 1992.
- [5] KUO, WAY; PRASAD V.RAJENDRA; TILLMAN, FRANK A.;Y HWANG CHING-LAI. **Optimal reliability design-Fundamentals and applications**.Cambridge University Press, 2001.
- [6] KUO, WAY Y ZUO, Ming.J. **Optimal reliability modeling. Principles and applications**. John Wiley & Sons, New Jersey, 2003.
- [7] CARSON, YOLANDA; MARIA, ANU. **Simulation Optimization: Methods and applications**. Proceedings of the IEEE Winter Simulation Conference. Ed. S.Andradóttir, K.J.Healy, D.H.Withers and B.L.Nelson, 1997.
- [8] BOWDEN, R.O.; HALL, J.D. **Simulation optimization research and development**. Simulation Conference Proceedings, 1998. Winter.Volume 2, 13-16 Dec. 1998 Page(s):1693 - 1698 vol.2
- [9] FARHAD AZADIVAR AND GEORGE TOMPKINS. **Simulation optimization with qualitative variables and structural model changes: A genetic algorithm approach** • European Journal of Operational Research, Volume 113, Issue 1, 16 February 1999, Pages 169-182
- [10] AZADIVAR, FARHAD. **Simulation Optimization Methodologies**. Proceedings of the IEEE Winter Simulation Conference. Paginas 93 a 100, 1999.
- [11] OLAFSSON, S.; JUMI KIM. **Towards a framework for black-box simulation optimization**. Simulation Conference, 2001. Proceedings of the WinterVolume 1, 9-12 Dec. 2001 Page(s):300 - 306 vol.1
- [12] SWISHER, J.R.; HYDEN, P.D.; JACOBSON, S.H.; SCHRUBEN, L.W.; **A survey of simulation optimization techniques and procedures**. Simulation Conference Proceedings, 2000. Winter_Volume 1, 10-13 Dec. 2000 Page(s):119 - 128 vol.1
- [13] ÓLAFSSON, SIGURDUR; KIM, JUMI. **Simulation Optimization**. Proceedings of the IEEE Winter Simulation Conference. Pages 79 - 82, 2002.
- [14] RUDOLPH, GÜNTER. **Finite Markov chain results in evolutionary computation: a tour d’horizon**. Disponible en: s11-www.cs.uni-dortmund.de/people/rudolph/publications/papers/FI.pdf –Consultado en: 28/02/2008
- [15] KURI, ANGEL FERNANDO E GALAVIZ, JOSÉ. **Algoritmos genéticos** D.R. Instituto Politécnico Nacional, Universidade Nacional Autônoma do México e Fundo de Cultura Econômica (UNAM), 1 ed.2002
- [16] DEB, KALYANMOY, PRATAP, AMRIT, AGARWAL, SAMEER & MEYARIVAN T. **A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II**. IEEE-Transactions on Evolutionary Computation, vol6, no.2, 2002.
- [17] ERGOTT, MATHIAS; GANDIBLEUX, XAVIER. Approximate Solution Methods for multiobjective Combinatorial Optimization. Sociedad de Estadística y Investigación Operativa. Madrid, Spain TOP Volume 12, number 1, 1-90 June 2004. Disponible en: <http://top.umh.es/top12101.pdf>. Acceso:07/02/2007

- [18] ZHEN XUE HAN; XU, L.; REN WEI; BO PING WANG; REINIKAMEN, T.; **Reliability-based design optimization for land grid array solder joints under thermo-mechanical load.** Thermal and Mechanical Simulation and Experiments in Microelectronics and Microsystems, 2004. EuroSimE 2004. Proceedings of the 5th International Conference on 2004 Page(s):219 – 224
- [19] XIANZHANG LEI; LERCH, E.; POVH, D.; KULICKE, B.; **Optimization-a new tool in a simulation program system [for power networks]** Power Systems, IEEE Transactions on Volume 12, Issue 2, May 1997 Page(s):598 - 604
- [20] TAN, K.C.; LEE, T.H.; KHOO, D.; KHOR, E.F.; **A multiobjective evolutionary algorithm toolbox for computer-aided multiobjective optimization.** Systems, Man and Cybernetics, Part B, IEEE Transactions on. Volume 31, Issue 4, Aug. 2001 Page(s):537 - 556
- [21] GULPMAR, NALAN; RUSTEM BERG; SETTERGEN, REUBEN. Simulation and optimization approaches to scenario tree generation. Journal of Economics Dynamics & Control 28 , 1291-1315, 2004.
- [22] CHEN SONG; XIAOHONG GUAN; QIANCHUAN ZHAO; YU-CHI HO; **Machine learning approach for determining feasible plans of a remanufacturing system.** Automation Science and Engineering, IEEE Transactions on [see also Robotics and Automation, IEEE Transactions on]. Volume 2, Issue 3, July 2005 Page(s):262 - 275
- [23] ROBIN, F.; ORZATI, A.; MORENO, E.; HOMAN, O.J.; BACHTOLD, W.; **Simulation and evolutionary optimization of electron-beam lithography with genetic and simplex-downhill algorithms.** Evolutionary Computation, IEEE Transactions on Volume 7, Issue 1, Feb. 2003 Page(s):69 - 82
- [24] BERTILSSON, K.; NILSSON, H.-E.; **The power of using automatic device optimization, based on iterative device simulations, in design of high-performance devices** Semiconductor Device Research Symposium, 2003 International. 10-12 Dec. 2003 Page(s):152 - 153
- [25] ROSEN, SCOTT L., HARMONOSKY, CATHERINE M. AND TRABAND, MARK T. A **simulation optimization** method that considers uncertainty and multiple performance measures European Journal of Operational Research, In Press, Corrected Proof, Available online 9 August 2006,
- [26] YANG, TAHO AND CHOU, POHUNG. Solving a multiresponse **simulation-optimization** problem with discrete variables using a multiple-attribute decision-making method. Mathematics and Computers in Simulation, Volume 68, Issue 1, 3 February 2005, Pages 9-21
- [27] GUASCH, ANTONI; PIERA, MIGUEL ANGEL; CASANOVAS, JOSEP; FIGUERAS, JAUME. **Modelado y simulación. Aplicación a procesos logísticos de fabricación y servicios.** Alfaomega, México DF, 2005.
- [28] SUBBU, R.; GOEBEL, K.; FREDERICK, D.K.; **Evolutionary design and optimization of aircraft engine controllers.** Systems, Man and Cybernetics, Part C, IEEE Transactions on Volume 35, Issue 4, Nov. 2005 Page(s):554 - 565
- [29] GUARISO, GIORGIO; HITZ, MARTIN; WERTHNER, HANNES. **An integrated simulation and optimization modeling environment for decision support.** Decision Support Systems 16, 103-117, 1996.
- [30] GASTON, G.J.; WALTON, A.J.; **The integration of simulation and response surface methodology for the optimization of IC processes.** Semiconductor Manufacturing, IEEE Transactions on Volume 7, Issue 1, Feb. 1994 Page(s):22 - 33

- [31] ABSPOEL, S.J.; ETMAN, L.F.P.; VERVOORT, J.; ROODA, J.E.;
Simulation optimization of stochastic systems with integer variables by sequential linearization. Simulation Conference Proceedings, 2000. Winter Volume 1, 10-13 Dec. 2000 Page(s):715 - 723 vol.1
- [32] CONTRERAS, J.; LOSI, A.; RUSSO, M.; WU, F.F.;**Simulation and evaluation of optimization problem solutions in distributed energy management systems.**Power Systems, IEEE Transactions on Volume 17, Issue 1, Feb. 2002 Page(s):57 - 62
- [33] DA-PENG WU; MING LU; JIAN-PING ZHANG; **Efficient optimization procedures for stochastic simulation systems.** Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on. Volume 5, 18-21 Aug. 2005 Page(s):2895 - 2900 Vol. 5
- [33] GONDA NEDDERMEIJER, H.; VAN OORTMARSSSEN, G.J.; PIERSMA, N.; DEKKER, R.;**A framework for Response Surface Methodology for simulation optimization.** Simulation Conference Proceedings, 2000. Winter. Volume 1, 10-13 Dec. 2000 Page(s):129 - 136 vol.1
- [34] JELONEK, T.M.; REILLY, J.P.; WU, Q.;**Real-Tiempo analog global optimization with constraints: application to the direction of arrival estimation problem.**Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on [see also Circuits and Systems II: Express Briefs, IEEE Transactions on]Volume 42, Issue 4, April 1995 Page(s):233 - 244
- [35] BARROS, M.; SILVA, J.; NEVES, G.; HORTA, N.; **Enhanced genetic algorithm kernel applied to a circuit-level optimization E-design environment.** [Electronics, Circuits and Systems, 2003. ICECS 2003. Proceedings of the 2003 10th IEEE International Conference on.](#) Volume 3, 14-17 Dec. 2003 Page(s):1046 - 1049 Vol.3
- [36] ZHANG, L.B.; ZHOU, C.G.; LIU, X.H.; MA, Z.Q.; MA, M.; LIANG, Y.C.; **Solving multi objective optimization problems using particle swarm optimization.** Evolutionary Computation, 2003. CEC '03. The 2003 Congress on. Volume 4, 8-12 Dec. 2003 Page(s):2400 - 2405 Vol.4
- [37] ALKHAMIS, TALAL M. AND AHMED, MOHAMED A. **A modified Hooke and Jeeves algorithm with likelihood ratio performance extrapolation for simulation optimization** European Journal of Operational Research, Volume 174, Issue 3, 1 November 2006, Pages 1802-1815
- [38] BAESLER, FELIPE F.; SEPÚLVEDA, JOSE A. **Multi-response simulation optimization using stochastic genetic search within a goal programming framework.** Proceedings of the IEEE Winter Simulation Conference. J.A.Joines, R.R.Barton, K.Kang y P.A.Fishwick eds., 2000.
- [39] TROCHU FRANÇOIS, RUIZ EDU, ACHIM VINCENT AND SOUKANE SOFIANE **Advanced numerical simulation of liquid composite molding for process analysis and optimization.****Composites Part A: Applied Science and Manufacturing**, Volume 37, Issue 6, June 2006, Pages 890-902
- [40] SEYED HAMID REZA PASANDIDEH AND SEYED TAGHI AKHAVAN NIAKI. **Multi-response simulation optimization using genetic algorithm within desirability function framework.** Applied Mathematics and Computation, Volume 175, Issue 1, 1 April 2006, Pages 366-382
- [41] **Simulation and optimization** of an uneven parallel bar release-regrasp maneuver Journal of Biomechanics, Volume 39, Supplement 1, 2006, Pages S180-S181 A.L. Sheets and M. Hubbard

- [42] TOMPKINS, GEORGE, AZADIVAR, FARHAD. **Genetic Algorithms in optimizing simulated systems.** Proceedings of the IEEE Winter Simulation Conference. Ed. Alexopoulos & Otros, 1995.
- [43] ROSEN, SCOTT L. AND HARMONOSKY, CATHERINE M. **An improved simulated annealing simulation optimization method for discrete parameter stochastic systems.** Computers & Operations Research, Volume 32, Issue 2, February 2005, Pages 343-358
- [44] KENNE P. AND GHARBI A. **A simulation optimization based control policy for Falla prone one-machine, two-product manufacturing systems.** Computers & Industrial Engineering, Volume 46, Issue 2, April 2004, Pages 285-292 J.
- [45] NAVA PLISKIN AND SOSNA, DAVID EBEN-CHAIME, MOSHE. **An integrated architecture for simulation.** Computers & Industrial Engineering, Volume 46, Issue 1, March 2004, Pages 159-170
- [46] IASSINOVSKI, A. ARTIBA, V. BACHELET AND F. RIANE **Integration of simulation and optimization for solving complex decision making problems.** International Journal of Production Economics, Volume 85, Issue 1, 11 July 2003, Pages 3-10 S.
- [47] AHMED, MOHAMED A. AND ALKHAMIS, TALAL M.. **Simulation-based optimization using simulated annealing with ranking and selection.** Computers & Operations Research, Volume 29, Issue 4, April 2002, Pages 387-402
- [48] MEDAGLIA, ANDRÉS L., SHU-CHERNG FANG AND NUTTLE, HENRY L. W. **Fuzzy controlled simulation optimization.** Fuzzy Sets and Systems, Volume 127, Issue 1, 1 April 2002, Pages 65-84
- [48] LEE, HYUN-MOON SHIN AND BYUNG-HEE YANG. **An approach for multiple criteria simulation optimization with application to turning operation.** Computers & Industrial Engineering, Volume 30, Issue 3, July 1996, Pages 375-386 Young-Hae
- [49] RUDOLPH, GÜNTER - AGAPIE, ALEXANDRU. **Convergence properties of some multiobjective evolutionary algorithms.** Proceedings of 2000 Congress of Evolutionary Computation. IEEE.2000
- [51] LUCE, K.; TREPANIER, L.; CIOCHETTO, F.; GOLDMAN, L.; **Simulation and optimization as effective DFSS tools.** Winter Simulation Conference, 2005 Proceedings of the.4-7 Dec. 2005 Page(s):7 pp.
- [52] GLOVER, F.; KELLY, J.P.; LAGUNA, M.; **New advances for wedding optimization and simulation.** Simulation Conference Proceedings, 1999. Winter Volume 1, 5-8 Dec. 1999 Page(s):255 - 260 vol.1
- [53] HICKS, D.A.; **A four step methodology for using simulation and optimization technologies in strategic supply chain planning.** Simulation Conference Proceedings, 1999. Winter. Volume 2, 5-8 Dec. 1999 Page(s):1215 - 1220 vol.2
- [54] ERBAS, C.; ERBAS, S.C.; PIMENTEL, A.D.; **A multiobjective optimization model for exploring multiprocessor mappings of process networks.** Hardware/Software Co design and System Synthesis, 2003. First IEEE/ACM/IFIP International Conference on 1-3 Oct. 2003 Page(s):182 – 187
- [55] PRICE, R.N.; HARRELL, C.R.; **Simulation modeling and optimization using ProModel.** Simulation Conference Proceedings, 1999. Winter Volume 1, 5-8 Dec. 1999 Page(s):208 - 214 vol.1
- [56] LOOP, B.P.; SUDHOFF, S.D.; ZAK, S.H.; ZIVI, E.L.; **An optimization approach to estimating stability regions using genetic algorithms.** American Control Conference, 2005. Proceedings of the 2005 8-10 June 2005 Page(s):231 - 236 vol. 1

- [57] FRERICK, CONSTANTIN, KREIS, PETER, GÓRAK, ANDRZEJ AND MELZNER, DIETER **Simulation and optimisation of the downstream process for purification of human serum albumin by using ion exchange membrane adsorbers** • Desalination, Volume 200, Issues 1-3, 20 November 2006, Pages 468-469
- [58] LAGANÁ ,DEMETRIO, LEGATO, PASQUALE, PISACANE, ORNELLA AND VOCATURO, FRANCESCA Solving **simulation optimization** problems on grid computing systems • Parallel Computing, Volume 32, Issue 9, October 2006, Pages 688-700
- [58] KOYAMADA, K.; SAKI, K.; ITOCH, T.;**Parameter optimization technique using the response surface methodology**. Engineering in Medicine and Biology Society, 2004. EMBC 2004. Conference Proceedings. 26th Annual International Conference of the. Volume 2, 2004 Page(s):2909 - 2912 Vol.4
- [59] HILTCHER, YANN, GUINGAND, MICHÈLE AND DE VAUJANY, JEAN PIERRE. Numerical **simulation and optimisation of worm gear cutting**. Mechanism and Machine Theory, Volume 41, Issue 9, September 2006, Pages 1090-1110
- [60] KHATTAB, N.M; **Simulation and optimization of a novel solar-powered adsorption refrigeration module**. Solar Energy 80, 823-833. 2006.
- [61] LAN H., OU J., ARMSTRONG, C. G. AND PRICE, M. A. An FE **simulation and optimisation** approach for the forging of aeroengine components .Journal of Materials Processing Technology, Volume 151, Issues 1-3, 1 September 2004, Pages 208-216
- [62] YAO. On constrained **simulation and optimization** by Metropolis chains. Statistics & Probability Letters, Volume 46, Issue 2, 15 January 2000, Pages 187-193 J.
- [63] TALAL M. ALKHAMIS, AHMED, MOHAMED A. AND TUAN, VU KIM. **Simulated annealing for discrete optimization with estimation**. European Journal of Operational Research, Volume 116, Issue 3, 1 August 1999, Pages 530-544
- [64] HUANG W. AND LAM H.N. **Using genetic algorithms to optimize controller parameters for HVAC systems**. Energy and Buildings, Volume 26, Issue 3, 1997, Pages 277-282
- [65] BINDER, T.; HEITZINGER, C.; SELBERHERR, S.;**A study on global and local optimization techniques for TCAD analysis tasks** .Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on Volume 23, Issue 6, June 2004 Page(s):814 - 822
- [66] POWELL, W.B.; **Optimization models and algorithms: an emerging technology for the motor carrier industry**. Vehicular Technology, IEEE Transactions on Volume 40, Issue 1, Part 2, Feb. 1991 Page(s):68 - 80
- [67] MAGOULAS, G.D.; ELDABI, T.; PAUL, R.J.; **Global search strategies for simulation optimization**. Simulation Conference, 2002. Proceedings of the Winter Volume 2, 8-11 Dec. 2002 Page(s):1978 - 1985 vol.2
- [68] CONG, J.; LEI HE; **Theory and algorithm of local-refinement-based optimization with application to device and interconnect sizing**. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on Volume 18, Issue 4, April 1999 Page(s):406 - 420
- [69] CUSICK, T. W. **Airfield simulation with optimization via decomposition** Mathematical and Computer Modeling, Volume 39, Issues 6-8, March 2004, Pages 745-758

- [70] KEYS, ANTHONY C.; REES, LOREN PAUL. **A sequential-design metamodeling strategy for simulation optimization.** Computers & Operations Research, 31, pages 1911-1932, 2004.
- [71] HONGWEI DING, LYÈS BENYOUCEF AND XIAOLAN XIE. **A simulation-based multi-objective genetic algorithm approach for networked enterprises optimization.** Engineering Applications of Artificial Intelligence, Volume 19, Issue 6, September 2006, Pages 609-623
- [72] SOARES BARATA, C. GUEDES, MARSEGUERRA M. AND ZIO E.- **Simulation modeling of repairable multi-component deteriorating systems for 'on condition' maintenance optimization.** Reliability Engineering & System Safety, Volume 76, Issue 3, June 2002, Pages 255-264 J.
- [73] CHUN-HUNG CHEN; WU, S.D.; DAI, L.; **Ordinal comparison of heuristic algorithms using stochastic optimization.** Robotics and Automation, IEEE Transactions on Volume 15, Issue 1, Feb. 1999 Page(s):44 - 56
- [75] WANG, LING. **A hybrid genetic algorithm–neural network strategy for simulation optimization.** Applied Mathematics and Computation, Volume 170, Issue 2, 15 November 2005, Pages 1329-1343
- [75] MOLLAGHASEMI, M.; EVANS, G.W.; **Multicriteria design of manufacturing systems through simulation optimization.** Systems, Man and Cybernetics, IEEE Transactions on Volume 24, Issue 9, Sept. 1994 Page(s):1407 - 1411
- [76] DILAY, E., VARGAS, J.V.C., AMICO, S.C. AND ORDONEZ, J.C. **Modeling, simulation and optimization of a beer pasteurization tunnel** • Journal of Food Engineering, Volume 77, Issue 3, December 2006, Pages 500-513
- [78] GEN, MITSUO and CHENG, RUNWEI. Genetic Algorithms & Engineering Optimization. Canada, John Wiley Sons. A Wiley-interscience publication, 2000.
- [79] DEB, KALYANMOY. **Multi-objective Optimization using Evolutionary Algorithms.** John Wiley & Sons, LTD, England, 2001.
- [79] ALLAOUI H. AND ARTIBA, A. **Integrating simulation and optimization to schedule a hybrid flow shop with maintenance constraints** .Computers & Industrial Engineering, Volume 47, Issue 4, December 2004, Pages 431-450.
- [80] NESMACHNOW, SERGIO. **Una versión paralela del algoritmo evolutivo para optimización multiobjetivo NSGA II y su aplicación el diseño de redes de comunicaciones confiables.** Disponible en: <http://www.fing.edu.uy/~sergion/MOEA/>. Consultado en: 13/01/2008
- [81] ZITZLER, ECKART, THIELE, LOTHAR; LAUMANNNS, MARCO. **SPEA-2: Improving the Strength Pareto Evolutionary Algorithm.** Technical Report TIK-Rep Switzerland, 2001. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.112.5073&rep=rep1&type=pdf>
Date:22/01/10
- [82] LAUMANNNS, MARCO; THIELE, LOTHAR; DE, KALYANMOY; ZITZLER, ECKART. **Combining convergence and diversity in evolutionary multi-objective optimization.** Massachusetts, Evolutionary computation, vol.10, no 3, 2002
- [83] ROBIC, TEA; FILIPIC, BOGDAN. **DEMO: Differential Evolution for Multiobjective Optimization.** Available at: citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.107.6104... Date:21/01/2010
- [84] GULIACHKI, VASSIL; TOSHEV, HRISTO; KORSEMOV, CHAVDAR. **Survey of evolutionary algorithms used in multiobjective optimization.** Bulgarian Academy of Sciences, Problems of engineering cybernetics and robotics, 60 – Sofia, 2009

- [85] SCHRUM, JACOB. **M 358K:Statistic Project Progress of the Evolutionary algorithm NSGA-II.** Available at:
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.140.2382>.
 Date:21/01/2010.
- [86] MUKERJEE, AMITABHA; BISWAS, RITA; DEB, KLYANMOY & MATHUR AMRIT P. **Multiobjective Evolutionary Algorithms for the Risk-return trade-off in Bank Loan Management.** *International Transactions in Operational Research*, volume 9, Issue 5, pages 583-597, 2002.
- [87] BROCKHOFF, DIMO; ZITZLER, ECKART. **Objective Reduction in Evolutionary Multiobjective Optimization: Theory and Applications.** *Evolutionary Computation* 17(2) pages 135-166, Massachusetts Institute of Technology, 2009.
- [88] TOSHEV, HRISTO; KORSEMOV, CHAVDAR. **Development of the evolutionary Approaches in multiobjetivo optimization.** *Sofia, Bulgarian Academy of Sciences, Problems of Engineering Cybernetics and Robotics*, 58, 2007.
- [89] ESKANDARI, HAMIDREZA; RABELO, LUIS; MOLLAGHASEMI, MANSOOREH. **Multiobjective simulation optimization using enhanced genetic algorithm.** *Proceedings of the 2005 Winter Simulation Conference*, 2005.
- [90] Toscano Pulido, Gregorio and Coello Coello, Carlos A. **“The Micro Genetic Algorithm 2: Towards On-Line Adaptation in Evolutionary Multiobjective Optimization”**, in Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb and Lothar Thiele (Eds), *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pp. 252–266, Springer, Lecture Notes in Computer Science, Vol. 2632, Faro, Portugal, April 2003.
- [91] REYNOLDS, ALAN P; DE LA IGLESIA BEATRIZ. **A Multi-Objective GRASP for Partial Classification.** Springer, *Soft Computing – A Fusion of Foundations, Methodologies and Applications*, 2008
- [92] JADAAN, OMAR; RAJAMANI, LAKISHMI; RAO, C.R. **Non-dominated ranked genetic algorithm for solving multi-objective optimization problems: NRGGA.** *Journal of theoretical and applied information technology*, JATIT, 2008.
- [93] ZITZLER, ECKART, THIELE, LOTHAR; BADER, JOHANNES. **SPAM:Set preference algorithm for multiobjective optimization.** Springer Berlin/Heidelberg, *Lectures Notes in Computer Science*, vol. 5199, p 847-858, 2008.
- [94] BECHIKH, SLIM; BELGASMI, NABIL; BEN SAID, LAMJED; GHIDIRA, KHALED; **PHC-NSGA-II: A Novel Multi-objective Memetic Algorithm for Continous Optimization.** *Proceedings of 20th IEEE International Conference on Tools with Artificial Intelligence*, 2008
- [95] SCHUETZE, OLIVER; SANCHEZ, GUSTAVO; COELLO, CARLOS A.COELLO. **A new memetic strategy for the numerical treatment of multi-objetivo optimization problems.** *Proceedings of the 10th annual conference on Genetic and evolutionay computation*, 2008
- [96] HU, JIANJUN; SEO, KISUNG; FAN, ZHUN AND OTHERS. **HEMO: A sustainable Multi-objective evolutionary optimization framework.** Springer Berlin/Heidelberg, *Lecture Notes in Computer Science*, vol 2723, /2003
- [97] KULVANIT, PASAN; PIROONRATANA, THEERA AND OTHERS. **Evolutionary multiobjective optimization by diversity control (MODCGA-II).** Springer Berlin/Heidelberg, *Lecture Notes in Computer Science*, vol 3967, 2006.
- [98] WANG, XUESONG; HAO, MINGLIN; CHENG, YUHU. **PDE-PEDA: A new pareto-based multi-objective optimization algorithm.** *Journal of Universal Computer Science*, vol 15, no.4, 722-741, 2009

- [99] SHARMA, DEEPAK; KUMAR, ABHAY; DEB, KALYANMOY; SINDHYA, KARTHIK. **Hybridization of SBX based NSGA-II and sequential quadratic programming for solving multi-objective optimization problems.** Available at: www.iitk.ac.in/kangal/papers/k2007007.pdf. On 21/01/2010.
- [100] DEB, KALYANMOY, LELE, SWANAND; DATTA RITUPARNA. **A hybrid evolutionary multi-objective and SQP base procedure for constrained optimization.** Springer Berlin/Heidelberg, Lectures Notes in Computer Science, volume 4683, 2007.
- [101] GAO, XIAODAN; CHEN, BINGZHEN; HE, XIARONG; QIU, Tong and others. **Multi-objective optimization for the periodic operation of the naphtha pyrolysis process using a new parallel hybrid algorithm combining NSGA-II with SQP.** Computer & Chemical Engineering, Volume 32, Issue 11, pages 2801-2811, 2008
- [102] BHUTANI, NAVEEN; RAY, AJAY K.; RANGAIAH G.P. **Modeling, simulation, and multi-objective optimization of an industrial hydrocracking unit.** Industrial & Engineering chemistry Research, 45 (4), pp 1354-1372, 2006.
- [103] GREINER. D; WINTER. G, EMPERADOR. J.M, GALVÁN, B. **A comparative analysis of “controlle elitism” in the NSGA-II applied to frame optimization.** Proceedings of the IUTAM Symposium on Evolutionary Methods in Mechanics, 101-11, 2004
- [104] ISHIBUCHI, HISAO; NARUKAWA, KANAME. **Comparison of Evolutionary multiobjective optimization with reference solution based single objective approach.** Proceedings of the 2005 conference on Genetic and evolutionary computation, 2005.
- [105] DEB, KALYANMOY; GOEL, TUSHAR. **A hybrid multi-objective evolutionary approach to engineering shape design.** Proceedings of the First International Conference on Evolutionary Multicriterion Optimization EMO, 2001.
- [106] NAIN, PAWAN K.S.; DEB, KALYANMOY. A multi-objective optimization procedure with successive approximate models. Available at: www.iitk.ac.in/kangal/papers/k2004012.pdf. Date:22/01/2010
- [107] AGGARWAL, VARUN; O'REILLY UNA-MAY. **COSMO: A correlation Sensitive mutation operator multi-objective optimization.** Proceedings of the 9th annual conference on Genetic and evolutionary computation, 2007
- [108] MATEO, PEDRO M.; ALBERTO ISOLINA. **Re-implementing NSGA-II and SPEA-II using operators based on Pareto ranking.** The Pyrenees International Workshop on Statistics, Probability and Operational Research, SPO, Jaca, 2009.
- [108] DEB, KALYANMOY; KARTHIK, S; OKABE, TATSUYA. **Self-adaptive Simulated Binary Crossover for real-parameter optimization.** Proceedings of the 9th annual conference on Genetic and evolutionary computation, 2007
- [109] JENSEN, MIKKEL T. **Reducing the Run-Time complexity of multiobjective EAs: The NSGA-II and other algorithms.** IEEE Transactions on Evolutionary computation, vol.7, no 5, 2003.
- [110] DEB, KLYANMOY; PADMANBHAN, DANESH AND OTHERS. **Handling uncertainties through reliability-based optimization using evolutionary algorithms.** Proceedings of the EMO Conference, 2007. Reliable on: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.135.9662&rep=rep1&type=pdf>. Date:21/01/2010
- [111] COELLO, CARLOS A. COELLO. **Evolutionary Multiobjective Optimization after 20 years: Current state and future.** Seminar on new trends on Intelligent Systems and Soft Computing, february 16th, 17th 2006. Available at: sci2s.ugr.es/seminars/4/talk-granada2006.pdf. Date:02.02.2010

- [112] Open Beagle W3. C++ Evolutionary Computation framework. Reliable at: <http://beagle.gel.ulaval.ca/>. Date: 22/01/2010
- [113] Paradiseo. A white-box object-oriented framework dedicated to the flexible design of metaheuristics. Available at: <http://paradiseo.gforge.inria.fr>. Date:22/10/2010.
- [114] DEME – Distributed Metaheuristics. The DEME project is devoted to the study of multi-objective optimization problem solving by using sequential, parallel, and distributed metaheuristic algorithms. Available at: <http://neo.lcc.uma.es/software/deme/index.php>. Date:22/01/2010
- [115] OptiGen Library – Genetic Software. Available at: <http://www.nd.com/genetic/>. Date: 22/01/2010
- [116] SINGLE AND MULTIOBJECTIVE GENETIC ALGORITHM TOOLBOX IN C++. Available at: <http://www.kumarasastry.com/2007/06/11/single-and-multiobjective-genetic-algorithm-toolbox-in-c/>. Date:22/01/2010
- [117] Rozenberg. Grzegorz; Reisig Wolfgang; Desel Jörg. **Lectures on concurrency and Petri Nets. Advances in Petri Nets.** Lecture notes in Computer Science. Vol. 3098. 2004. DOI 10.1007/b98282
- [118] Murata. T. **Petri Nets: Properties. Analysis and applications.** Proceedings of the IEEE. Vol. 77. No 4. April 1989. pp 541-580.
- [119] Peterson. J.L. **Petri Net Theory and the Modeling of Systems.** Prentice-Hall. N.J. 1981.
- [120] Petri Nets World. <http://www.informatik.uni-hamburg.de/TGI/PetriNets/>
- [121] MISRA, KRISHNA B. AND SHARMA, USHA. **An efficient approach for multiple criteria redundancy optimization problems.** Microelectronics Reliability, volume 31, Issues 2/3, pp 303-321. doi:10.016/0026-2714(91)90216. 1991
- [122] JIANPING, LI AND XISHEN, JIA. **A new partial bound enumeration technique for solving reliability redundancy optimization.** Microelectronics Reliability, Volume 37, Issue 2, pp 237-242. Doi 10.1016/S0026-2714(96)00109-6. 1997
- [123] TIAN, SHIGANG AND SUO MING J. **Redundancy allocation for multi-state systems using physical programming and genetic algorithms.** Reliability Engineering and System Safety, Volume 91 pp 1049-1056. doi: 10.1016/j.ress.2005.11.039. 2006
- [124] KUMAR, RANJAN, IZUI, KAZUHIRO; YOSHIMURA, MASATAKA; AND NISHIWAKI, SHINJI. **Multi-objective hierarchical genetic algorithms for multilevel redundancy allocation optimization.** Reliability Engineering and System Safety. Volume 94, Issue 4, pp 891-904. doi: 10.016/j.ress.2008.10.002. 2009
- [125] WANG, ZAI; CHEN TIANSHI; TANG, KE AND YAO, XIN. **A Multi-objective Approach to redundancy allocation problem in parallel-series Systems.** IEEE Congress on Evolutionary Computation – CEC, pp 582-589. doi: 10.1109/CEC.2009.4982998. 2009
- [126] SHELOKAR, PRAKASH S.; JAYARAMAN, V.K.; KULKAMI, B.D. **Ant algorithm for single and multiobjective reliability optimization problems.** Quality and Reliability Engineering International, Volume 18, pp 497-514. doi: 10.1002/qre.499. 2002
- [127] KULTUREL-KONAK, SADAN; SMITH, ALICE E. AND NORMAN, BRYAN A. **Multi-objective tabu search using a multinomial probability mass function.**

- European Journal of Operational Research, Volume 169, pp 918-931. doi:10.1016/j.ejor.2004.08.026. 2006
- [128] TABOADA, HEIDI A. BAHERANWALA, FATEMA, COIT, DAVID W. AND WATTANAPONGSAKORN, NARUEMON . **Practical solutions for multi-objective optimization: An application to system reliability design problems.** Reliability Engineering and System Safety No 92. pp. 314-322. doi: 10.1016/j.ress.2006.04.014. 2007
- [129] COELHO, LEANDRO DOS SANTOS. **Reliability-redundancy optimization by means of a chaotic differential evolution approach.** Chaos, Solitons and Fractals, Volume 41, Issue 2, pp 594-602. doi: 10.1016/j.chaos.2008.02.028. 2009
- [130] MEEDENIYA, INDIKA; ALETI, ALDEIDA AND BUHNOVA, BABOVRAA . **Redundancy Allocation in Automotive systems using Multi-objective Optimization.** Jacobs School UCSD Symposium on Automotive/Avionics Systems Engineering SAASE 2009. http://www.jacobsschool.ucsd.edu/GordonCenter/g_leadership/l_summer/s_saase/schedule.shtml
- [131] LIANG, YUN-CHIA AND LO, MIN-HUA. **Multi-objective redundancy allocation optimization using a variable neighborhood search algorithm.** Journal of Heuristics, Volume 16, pp 511-535. doi:10.1007/s10732-009-9108-4. 2010
- [132] KHALILI-DAMGHANI, KAVEH AND AMIRI MAGHSOUD. **Solving binary-state multi-objective reliability redundancy allocation series-parallel problem using efficient epsilon-constraint, multi-start partial bound enumeration algorithm, and DEA.** Reliability Engineering and System Safety, Volume 103, pp 35-34. doi: 10.1016/j.ress.2012.03.006. 2012
- [133] SALAZAR, DANIEL; ROCCO, CLAUDIO M AND GALVÁN, BLAS J. **Optimization of constrained multiple-objective reliability problems using evolutionary algorithms.** Reliability Engineering and System Safety. Volume 91, Issue 9, pp 1057-1070. doi: 10.1016/j.ress.2005.11.040. 2006
- [134] TABOADA, HEIDI A. AND COIT, DAVID W. **Development of multiple objective genetic algorithm for solving reliability design allocation problems.** Proceedings of the 2008 Industrial Engineering Research Conference. J. Fowler and S. Masons. eds. Available at: http://ie.rutgers.edu/resource/research_paper/paper_08-004.pdf. 2008
- [135] LINS, ISIS DIDIER AND DROGUETT, ENRIQUE LÓPEZ. **Redundancy allocation problems considering systems with imperfect repairs using multi-objective genetic algorithms and discrete event simulation.** Simulation Modelling Practice and Theory, volume 19, pp 362-381. doi: 10.1016/j.simpat.2010.07.010. 2011
- [136] TABOADA, HEIDI A. AND COIT, DAVID W. **A new multiple objective evolutionary algorithm for reliability optimization of series-parallel systems.** Applied Evolutionary Computation, Volume 3, Issue 2, pp 1-18. doi 10.4018/jaec.2012040101. 2012
- [137] REYES SIERRA, MARIA MARGARITA-COELLO COELLO, CARLOS A. **Tiempo Esperado de Convergencia de un algoritmo genético con parámetros mínimos.** CINVESTAV-IPN, México DF, 2003.

- [138] RUDOLPH, GÜNTER. **Convergence properties of canonical genetic algorithms.** IEEE Transactions on Neural Networks, 1(5):96-101, 1994.
- [139] RUDOLPH, GÜNTER - AGAPIE, ALEXANDRU. **Convergence properties of some multiobjective evolutionary algorithms.** Proceedings of 2000 Congress of Evolutionary Computation. IEEE.2000
- [140] WALPOLE, RONALD E.; MYERS, RAYMOND H. – **Probabilidad y estadística.** Mexico, McGraw-Hill/Interamericana de México S.A. de C.V, cuarta edición, 2004.
- [141] LAHOZ-BELTRA, RFAEL, PERALES-GRAVAN, CARLOS. **A survey of nonparametric tests for the statistical analysis of evolutionary computational experiments.** International Journal “Information Theories and Applications”, Vol.17, Number 1, p.49-61,2010.

ANEXO 1 – Modelo PETRI diagrama y tablas resumen

Tabla 19 Descripción de las operaciones del sistema

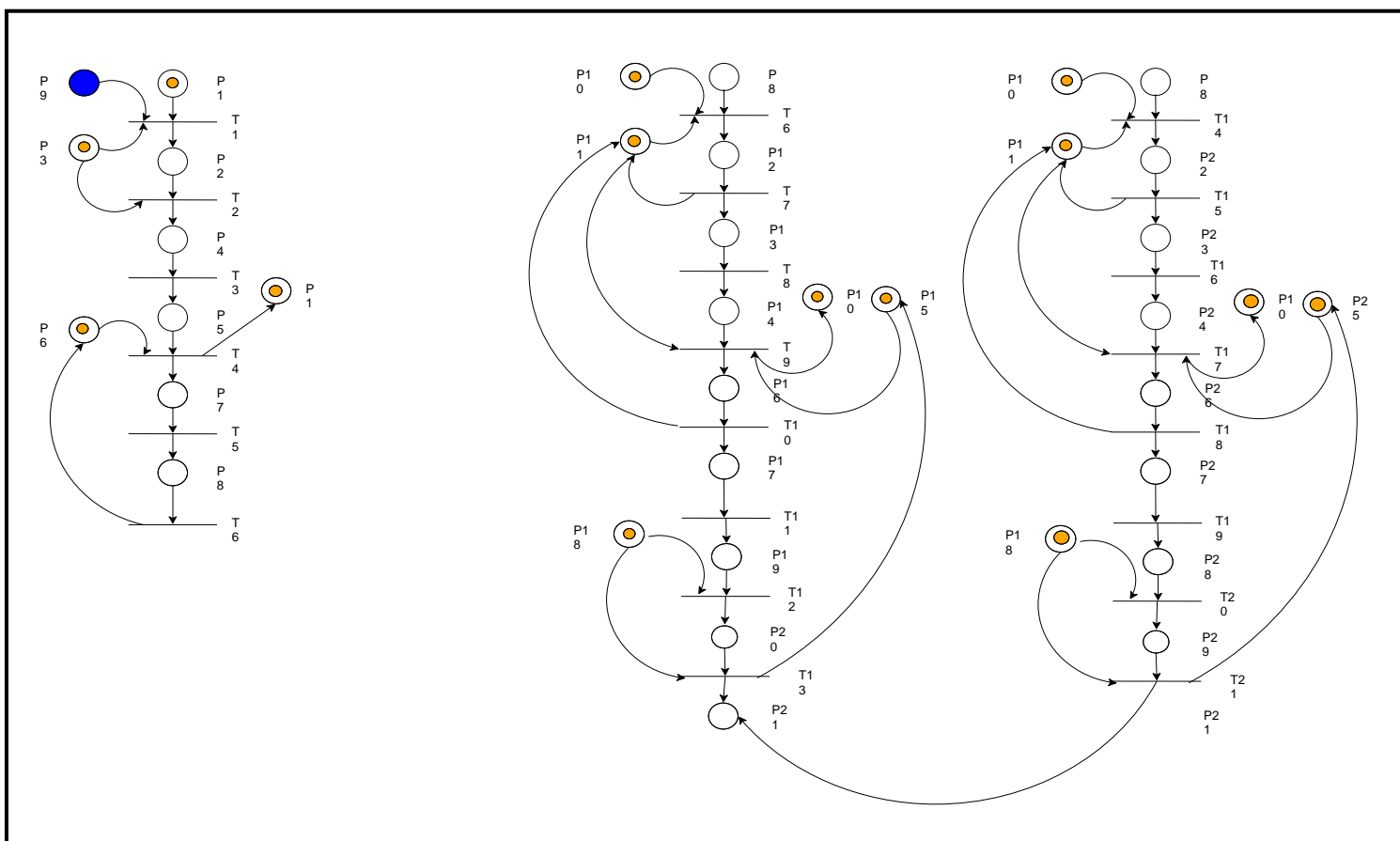
ítem	Transición	Descripción
01	T1	Inicio de la operación de transporte 1 de la pieza de S1 a M1
02	T2	Fin de la operación de transporte 1 e inicio del proceso de la máquina M1 – operación 1
03	T3	Fin de proceso de la máquina M1 - operación 1
04	T4	Inicio del proceso de erosión por la máquina M2 – operación 2
05	T5	Fin del proceso de erosión – operación 2
06	T6	Inicio de la operación de transporte 2 de la pieza de M2 a M3
07	T7	Fin de la operación de transporte 2 e inicio del proceso de ensamble – operación 3
08	T8	Fin del proceso de ensamble – operación 3
09	T9	Inicio de la operación de transporte 3 de la pieza de M3 a M5
10	T10	Fin de la operación de transporte e inicio del proceso de ajuste por la máquina M5 – operación 4
11	T11	Fin del proceso de ajuste – operación 4
12	T12	Inicio de la operación de transporte 4 de la pieza de M5 a S2(almacén final)
13	T13	Fin de la operación de transporte 4
14	T14	Inicio de la operación de transporte 5 de la pieza de M2 a M4
15	T15	Fin de la operación de transporte 2 e inicio del proceso de ensamble – operación 5
16	T16	Fin del proceso de ensamble – operación 5
17	T17	Inicio de la operación de transporte 6 de la pieza de M4 a M6
18	T18	Fin de la operación de transporte e inicio del proceso de ajuste por la máquina M6 – operación 6
19	T19	Fin del proceso de ajuste – operación 6
20	T20	Inicio de la operación de transporte 7 de la pieza de M6 a S2(almacén final)
21	T21	Fin de la operación de transporte 7

Tabla 20 Descripción de los nodos lugar de la RED de PETRI

Ítem	Lugar	Descripción
01	P ₁	Máquina M ₁ libre (Fresado)
02	P ₂	Robot R ₁ transportando pieza de S ₁ (almacén) a M ₁
03	P ₃	Robot R ₁ libre
04	P ₄	Máquina M ₁ procesando
05	P ₅	Máquina M ₁ ocupada con pieza ya procesada
06	P ₆	Máquina M ₂ (Erosión) libre
07	P ₇	Máquina M ₂ procesando
08	P ₈	Máquina M ₂ ocupada con pieza ya procesada
09	P ₉	Almacén S1: número de piezas en el stock de materia prima
10	P ₁₀	Máquina M3 libre (Ensamble A)
11	P ₁₁	Robot R ₂ libre
12	P ₁₂	Robot R ₂ transportando pieza de M ₂ para M ₃
13	P ₁₃	Máquina M ₃ procesando
14	P ₁₄	Máquina M ₃ ocupada con pieza ya procesada
15	P ₁₅	Máquina M ₅ libre
16	P ₁₆	Robot R ₂ transportando pieza de M ₃ para M ₅
17	P ₁₇	Máquina M ₅ procesando (Ajuste A)
18	P ₁₈	Robot R ₃ libre
19	P ₁₉	Máquina M ₅ ocupada con pieza ya procesada

20	P ₂₀	Robot R ₃ transportando pieza de M ₅ para almacén S2
21	P ₂₁	Almacén S ₂ : número de piezas en el stock de producto final
22	P ₂₂	Robot R ₂ transportando pieza de M ₂ para M ₄
23	P ₂₃	Máquina M ₄ procesando(Ensamble B)
24	P ₂₄	Máquina M ₄ ocupada con pieza ya procesada
25	P ₂₅	Máquina M ₆ libre
26	P ₂₆	Robot R ₂ transportando pieza de M ₄ para M ₆
27	P ₂₇	Máquina M ₆ procesando (Ajuste B)
28	P ₂₈	Máquina M ₆ ocupada con pieza ya procesada
29	P ₂₉	Robot R ₃ transportando pieza de M ₆ para almacén S2

Figura 13 Rede de Petri para el modelo conceptual del sistema



ANEXO 2 RESUMEN DE LOS ARTICULOS CONTENIDOS EN LAS REFERENCIAS.

Clasificación general de las referencias, basada en Smith según Bowden & Hall [19]. Para leer este anexo Primeramente se presenta los artículos numerados su posición en la referencia y agrupados por dominio en la secuela se presentan comentarios acerca de cada uno de los artículos listados

A. Dominio: Método

1. Multicriteria design of manufacturing systems through simulation optimization
2. An approach for multiple criteria simulation with application to turning operation
3. Using genetic algorithms to optimize controller parameters for HVAC
4. Simulation and evaluation of optimization problem solutions in distributed energy management systems
5. Simulation based optimization using simulated annealing with ranking and selection
6. An FE simulation and optimization approach for the forging of aero engine components
7. A simulation optimization based control policy for Falla one machine, two product manufacturing
8. Airfield simulation with optimization via decomposition
9. Simulation and optimization approaches to scenario tree generation
10. Solving a multiresponse simulation/optimization problem with discrete variables using a multiple-attribute decision making method
11. An improved simulate annealing simulation optimization method for discrete parameter stochastic systems
12. Simulation based sequencing and lot size optimization for a production and inventory system with multiple items
13. Solving simulation optimization problems on grid computing systems
14. Numerical simulation and optimization of worm gear cutting
15. A simulation optimization method that considers uncertainty and multiple performance measures
16. Advanced numerical simulation of liquid composite molding for process analysis and optimization
17. Dynamics of an activated sludge process with nitrification and denitrification. Star-up simulation and optimization using evolutionary algorithm
18. Simulation and optimization of a novel solar-powered absorption refrigeration module
19. Simulation modeling of repairable multi-component deteriorating systems for "on condition" maintenance optimization
20. A multiobjective optimization model for exploring multiprocessor mappings of process networks
21. Genetic algorithms in optimizing simulated systems
22. Theory and algorithm of local refinement based Optimization with application to device and Interconnect sizing
23. A four step methodology for using simulation and Optimization technologies in strategic supply chain
24. Planning
25. Parameter optimization technique using the response surface methodology
26. Interactive optimization and simulation of barge traffic on an inland waterway
27. Application of an ordinal optimization algorithm to the testing process
 - A.1 Comentarios acerca de los artículos clasificados en el dominio del método
 1. Presentan la técnica de optimización multicriterio llamada método STEP. Fue usado como ejemplo un sistema da manufactura
 2. Fue desarrollado un eficiente algoritmo multicriterio basado en la superficie de corte usando la metodología de la superficie de respuesta(RSM –Response Surface Methodology) para variables continuas
 3. Aplicación de la simulación/optimización para controlar el desempeño de un controlador PID de un sistema Calentamiento, ventilación y Aire acondicionado(HVAC en inglés)
 4. Aplicación de simulación optimización al gestión de sistemas de energía
 5. Presenta un método que combina recocido simulado con selección y tanteo para la resolución de problemas de optimización estocásticos.

6. Presenta un proceso de simulación basado en un método de elementos finitos y optimización aplicado a un proceso de fundición de componentes de turbinas de aviones
7. Una aplicación de la simulación optimización a un problema de escoja de las tasas de producción que minimizan los costos esperados del inventario sobre un horizonte infinito para un sistema de producción que contiene una máquina y dos productos.
8. El estudio describe el desarrollo y uso de un modelo computacional que analiza las complejas actividades que ocurren en un campo de aviación militar de manera a que permite que los campos de aviación tengan la habilidad de soportar muchos tipos de misiones para sus aeronaves. El simulador BRACE(Base Resource and Capabilities Estimator) fue acoplado al algoritmo optimizador basado en la metodología de descomposición.
9. El artículo presenta tres enfoques para la generación de árboles de escenarios para tres problemas de portafolios financieros a través de simulación/optimización basado en aproximación estocástica.
10. Propone un abordaje para la resolución de problemas no lineales y estocásticos pertinentes a los sistemas de producción resaltando que los problemas prácticos involucran muchas características de los problemas de optimización de múltiple respuesta y para la resolución de tales problemas proponen un métodos de toma de decisiones de múltiples atributos que es una técnica para ordenar preferencias por similitud a las soluciones ideales(TOPSIS del inglés technique for order preference by similarity to ideal solutions) El método asume que los factores de control tienen valores discretos y que cada factor de control tiene exactamente tres niveles de control. Funciones de pérdida de calidad de Taguchi fueran adaptadas al modelado del factor promedio y a los efectos de la variancia. Así se trata de una propuesta híbrida de los métodos de Taguchi y TOPSIS.
11. Los autores proponen un método de simulación optimización basado en el algoritmo recocido simulado desarrollado para mejorar el desempeño del recocido simulado para la simulación optimización de variables discretas, el método acopla la técnica de la superficie de respuesta al algoritmo recocido simulado
12. Un problema de decisión fue investigado considerando un sistema de manufactura como objeto de estudio y la necesidad de definirse políticas para el funcionamiento óptimo del sistema. Fue analizado el problema de definición de cual política de manufactura o las normas de secuenciación y las normas para el tamaño de lote las cuales maximizan las ganancias esperadas por unidad de tiempo. Una vez que el problema es muy complejo para las soluciones analíticas, los autores acotaran la búsqueda por una óptima política para simplificar la estructura de políticas, que puede ser descrita por pocos parámetros. Para encontrar los valores óptimos de los parámetros utilizaran simulación optimización dónde el simulador del sistema fue combinado con un algoritmo genético optimizador.
13. La óptima asignación de amarraderos y de grúas para los servicios marítimos es un problema logístico crucial para los modernos terminales de contenedores y deben ser formulados para tener en cuenta su naturaleza estocástica. Los autores usaran una rejilla computacional para resolver uno de los mayores problemas de los puertos a través del enfoque de optimización basado alrededor de un modelo de colas en red del proceso logístico de interés. Fue utilizado el algoritmo SARP que es una combinación del recocido simulado con el procedimiento de Rinotts para la selección de las soluciones generadas acoplada a un simulador de modelos de cola.
14. La fabricación de un mecanismo de tornillo sin fin requiere numerosos ajustes y los estándar de contacto son frecuentemente usados para validar la calidad del acoplamiento del mecanismo de tornillo sin fin. El artículo presenta una herramienta numérica que fue desarrollada para simular el acoplamiento del mecanismo y computar el error quine matico y los patrones de no contacto. La herramienta numérica es entonces usada para estudiar la influencia de un conjunto de parámetros de acoplamiento y formular un procedimiento de optimización de la producción antes de cualquier operación en la planta.
15. El amago de esta investigación es la incorporación a la simulación las preferencias de los usuarios finales de encuentro a los riscos e incertidumbres dentro del proceso de búsqueda de la mejor alternativa de decisión. Los autores llegan a la conclusión que para eso es necesario automatizar la optimización acoplada a la simulación a través de dos métodos de simulación optimización que contienen estrategias determinísticas de toma de decisiones múltiple criterio.
16. Los procesos de manufactura de compuestos basados en el moldeado de compuestos líquidos (LCM en inglés) he avanzado mucho en los recientes años. Enfocado en las etapas de llenado y cura durante el LCM; el trabajo presenta una suma de contribuciones en términos de los procesos de simulación y optimización que pueden ser concebidos para mejorar la eficiencia de la creación del prototipo virtual del molde.
17. El artículo presenta el modelado dinámico y el uso de la simulación del sistema de manera a establecer el modo de operación continua (steady state) de ambos los sistemas de nitrificación y desnitrificación. El sistema ASM3 fue aplicado para el modelado de los procesos microbiológicos en la

aeración del tanque y modelo de 10 capas de decantación es adaptado al tanque de decantación. Entonces, el problema de optimización es llevado por el algoritmo genético, y la óptima secuencia de aeración es determinada para del mínimo consumo de energía. El manejo de las restricciones es alcanzado por tres métodos de comparativos usando el rechazo de infactibilidades, la penalización de infactibilidades y la selección por ruleta sin cualquier parámetro de penalización.

18. Un modelo matemático fue desarrollado para simular optimizar el desempeño del módulo de refrigeración y absorción energizado por el sol que contiene dos tipos de sólidos de absorción domésticos el carbón y el metanol.
19. El trabajo presenta el modelado de un sistema degenerativo monitoreado continuamente por el uso de simulación Monte Carlo regido por un esquema de optimización del mantenimiento basado en la condición cuya finalidad es minimizar el costo total esperado del sistema para una dado tiempo de misión. El modelo de simulación es primeramente introducido por la consideración de una única componente no reparable sujeta a una degradación estocástica. El modelado es generalizado para un sistema reparable de múltiple componentes. Para encontrar los límites óptimos de degradación de una intervención del mantenimiento, el procedimiento de optimización del costo es hecho a través de una búsqueda simple en el espacio de los límites del mantenimiento (Método heurístico de optimización).
20. En el contexto del proyecto Artemis fue desarrollado el programa SESAME, que se trata de una estructura que provee métodos de simulación y modelado y herramientas para el diseño eficiente de la exploración de espacios de sistemas heterogéneos embebidos de multiTiempos acoplada a un algoritmo de optimización multiobjetivo llamado SPEA(descripción rápida en el anexo).
21. Este artículo es el prototipo de otros artículos de los autores en la área de optimización simulación destacan que muchas decisiones en el diseño involucran la escogida de políticas alternativas y frecuentemente las variables usadas para representar tales alternativas no son solamente discretas más cualitativas. Los autores presentan una metodología de simulación optimización que puede operar con variables cualitativas. Para eso proponen la unión de un algoritmo genético con un generador de modelos de simulación orientados por objetos. Los diseños del sistema recomendados por el AG son convertidos en modelos de simulación y ejecutados.
22. Aplicación en el área de desempeño de circuitos electrónicos. Tratase de un experimento que conjuga el método de optimización propuesto por Cong-He (Programa CH) y el algoritmo de computación acotada (bound computación) para la búsqueda de layout óptimo de un circuito.
23. Proponen una estrategia constituida de 4 etapas.- optimización-simulación-simulación/optimización para optimizar las actividades de planeación y
24. toma de decisiones en los sistemas de cadenas de suministro.
25. El artículo propone una técnica de optimización por el uso de la metodología de la superficie de respuesta (e Response Surface Methodology (RSM)) para la simulación exacta de células biológicas que calcula el comportamiento de la célula por integración numérica de la ecuación diferencial y genera acciones potenciales, transitorio celular Ca, contracción y consumo intracelular de ATP por el uso de los mayores valores de los parámetros de permeabilidad iónica y amplitud de la dependencia de K_o . Debido que el parámetro de espacio los autores no puede ser aproximado por una única superficie cuadrática polinomial, los autores adoptaran una técnica de subdivisión recursiva y el uso de coeficientes de múltiples determinaciones para representar la superficie de respuesta en cada sub-espacio para un dado criterio de subdivisión.
26. Proceso de simulación y optimización usando la combinación de un algoritmo recocido simulado + el heurísticos de Nawaz, Enscore & Ham, NEH acoplados a el simulador RAO(recursos, acciones, operaciones) para la resolución de un problema de asignación de una secuencia de fabricación sujeta a restricciones de mantenimiento.
27. El Hooke & Jeeves algoritmo (HJ) es un es un procedimiento estándar de búsqueda usado para optimizar funciones no lineales que no necesariamente son continuas y diferenciables. El algoritmo desempeña repetidamente dos tipos de rutinas de búsquedas: una búsqueda exploratoria y una búsqueda estándar. El algoritmo HJ requiere una evaluación determinística de la función a ser optimizada. Los autores consideraran situaciones dónde la función objetivo es estocástica y puede ser evaluada solamente a través de una simulación Monte Carlo. Para superar el problema del uso de extensivas evaluaciones por la simulación Monte Carlo la técnica de extrapolación del desempeño de la proporción de probabilidad.(LRPE en inglés). Ellos hicieron la extrapolación del desempeño medido para diferentes valores de los parámetros de decisión mientras simulaban un camino simple de muestreo del sistema fundamental. Su algoritmo modificado Hooke & Jeeves usó la extrapolación del desempeño de la proporción de probabilidad en el proceso de simulación y optimización.

B. Dominio de las Estrategias y tácticas

1. Simulated annealing for discrete optimization with estimation
2. Simulation optimization of stochastic systems with integer variables by sequential linearization
3. A framework for response surface methodology simulation optimization
4. On constrained simulation and optimization by Metropolis chains
5. Fuzzy controlled simulation optimization
6. Simulation and evolutionary optimization of electron-beam lithography with genetic and simplex. downhill algorithms
7. Integrating simulation and optimization to schedule a hybrid flow shop with maintenance constraints
8. A sequential-design metamodeling strategy for simulation optimization
9. Efficient optimization procedures for stochastic simulation systems
10. Machine learning approach for determining feasible plans of a remanufacturing system.
11. A hybrid genetic algorithm-neural network strategy for simulation
12. Modeling simulation and optimization of a beer pasteurization tunnel
13. A modified Hooke and Jeeves algorithm with likelihood ratio performance extrapolation for simulation optimization
14. Multi-response simulation optimization using genetic algorithm within desirability function framework
15. Multi-response simulation optimization using stochastic genetic search within a goal programming framework

B.1 Comentarios acerca del dominio de las estrategias y tácticas

1. Presentan el algoritmo recocido simulado modificado para un contexto optimización estocástica donde los cálculos de la función objetivo son perturbados por un ruido estocástico.
2. Fue desarrollada una estrategia de optimización basada en una serie de sub-problemas de aproximación lineal. Cada sub-problema es construido a partir del modelo de simulación para la resolución de problemas en los sistemas de manufactura caracterizados por unas funciones objetivo y de restricciones cuyos valores esperados son funciones estocásticas.
3. Desarrollo de una estructura automática para optimización de modelos de simulación estocásticos a través del método de superficies de respuesta, que según los autores es frecuentemente utilizada para la optimización de modelos de simulación estocástica de forma no automatizada.
4. Aplicación de cadenas de Metrópolis no homogéneas para simular o minimizar bajo algunas restricciones.
5. Presenta una estrategia de optimización que depende de un controlador fuzzy guiado por un conjunto de reglas derivadas de conceptos estadísticos, modelos de superficie de respuesta, y conocimiento de expertos para ilustrar la aplicación en un sistema de manufactura.
6. Programación lineal es usada como técnica de optimización conjugada a un modelo echo en el software ARENA. Es una técnica iterativa que fue aplicada a la optimización del tráfico de barcas en un canal.
7. Un abordaje híbrido usando un algoritmo genético para la búsqueda global y un algoritmo simple-downhill para una posible búsqueda local mejoro drásticamente la estructura optimizada solamente por el algoritmo genético. La estrategia fue aplicada a un cañón de electrones litografica para la fabricación de circuitos electrónicos de microondas
8. El artículo presenta la factibilidad y robustez de una estrategia secuencial de simulación/optimización usando un metamodelo no paramétrico, incorporando las nociones avanzadas de estadística no paramétrica en la literatura, utilizan el método "thin-plate spline sequential procedure"
9. Aplicación de la técnica de optimización enjambre de partículas (particle swarm optimization PSO) al modelo de simulación HKCONSIN al revés de un mecanismo convencional de simulación optimización (constante & promedio) aplican un proceso no constante y estocástico usando PSO lo comparan con un algoritmo genético.
10. Para la resolución de los problemas de planeación de las actividades de remanufactura a través de una nueva metodología basado en un enfoque de aprendizaje de máquina para predecir el plan requerido de viabilidad en aplicaciones prácticas. Por la aplicación de la teoría de los conjuntos aproximados para predecir las reglas que son obtenidas o aprendidas de un conjunto de datos de entrenamiento generado por el simulador. El simulador es acoplado a una estructura de optimización ordinal [33] y un nuevo método de optimización ordinal restringido fue desarrollado para la planeación de re manufactura. La planeación de recursos de sistemas complejos de re manufactura es generalmente extremadamente difícil en términos del tamaño del problema y sus incertidumbres, y, en muchos casos, un enfoque de simulación es único camino disponible. Pero la determinación previa de

un plan factible mejoraría en mucho la eficiencia de la simulación y provee un poderosa herramienta para futuros desarrollos de métodos de simulación para la optimización de la planeación de operaciones. El artículo presenta un problema de un sistema de re manufactura que fue formulado como un problema estocástico de optimización discreta con complicadas restricciones probabilísticas.

11. Fue presentada una técnica híbrida empleando un algoritmo genético acoplado a una red neuronal para la resolución del problema de optimización. El buen desempeño de aproximación de la red neuronal y la efectiva y robusta búsqueda evolutiva de los algoritmos genéticos son aplicadas en sentido híbrido donde la red neuronal es empleada en la predicción del valor de lo objetivo y el AG es adoptado en la búsqueda del diseño óptimo basado en los valores de aptitud (fitness) pronosticado. Una simulación numérica fue hecha para probar el método del bien conocido problema de proyecto de la vasija de presión ,
12. Para un problema de optimización estocástica para encontrar los valores límites que reducen las excesivas pérdidas bajo a niveles de re prueba tolerables en lo proceso de prueba de galletas. Según los autores es un problema de duro de optimización con un largo espacio de soluciones. Proponen entonces una teoría de optimización ordinal basada en un algoritmo de dos niveles para la resolución del vector de los valores de los límites suficientemente buenos y compara con los obtenidos por otros métodos usando un conjunto de 521 galletas de una prueba real.
13. Los autores introducen un modelo experimental de un túnel de pasteurización de cervezas que puede ser aplicado a cualquier túnel de la industria de alimentos. Un modelo matemático para simular computacional mente el comportamiento energético de túnel de pasteurización fue acoplado a un procedimiento de optimización que según los autores lograran con el método una combinación de la efectividad numérica con un bajo tiempo computacional.
14. El trabajo presenta una nueva metodología para resolver problemas de optimización estadística de múltiples respuestas. La metodología integra funciones deseables y el enfoque de simulación con un algoritmo genético. Las funciones deseables son responsables por el modelado del problema estadístico de múltiples respuestas, el enfoque de simulación genera los datos requeridos de entrada a partir del modelo del sistema para que finalmente el genético algoritmo intente optimizar el modelo. Esta metodología incluye 4 métodos. Cada método difiere uno del otro en el control de la aleatoriedad del problema. En el primero y segundo métodos, las replicaciones controlan esta aleatoriedad en el tercero y cuarto métodos ellos controlaran la variación por pruebas estadísticas.
15. El presenta, según sus autores, un nuevo enfoque para solución de problemas de múltiples respuestas de optimización y simulación por la integración de un modelo de simulación con un algoritmo genético y un modelo de programación por metas u objetivo (goal). El algoritmo genético ofrece una herramienta muy flexible y confiable para la búsqueda en un contexto global. Este método fue modificado para realizar una búsqueda considerando el promedio y la variancia de las respuestas. En esto camino, la búsqueda es realizada estocástica mente y no determinísticamente como en la mayoría de los enfoques reportados en la literatura de acuerdo con los autores.

C. Dominio de la Interfaz

1. Optimization models and algorithms: an emerging technology for the motor carrier industry
2. An integrated simulation and optimization modeling environment for decision support
3. Optimization a new tool in simulation program system for power networks
4. An integrated architecture for simulation
5. A simulation based multiobjective genetic algorithm approach for networked enterprises optimization

C1 – Comentarios acerca del dominio de la interfaz

1. El artículo está enfocado en el problema de optimización cargamento en tiempo real de camiones de compañías trasportadoras. Presentan el sistema grafico iterativo de simulación MIDAS que fue desarrollado para demostrar los métodos de optimización permitiendo al usuario recorrer diversos modelos de optimización en una estructura de tiempo real.
2. Desarrollo de un entorno de simulación basado en una estructura de orientada a objeto.
3. Presenta el entorno de simulación optimización NETMAC formulado a través de una lenguaje de simulación orientada a bloques
4. Propone una arquitectura de simulación/optimización que usa como interfaz un banco de datos.
5. Presentan una herramienta de simulación optimización basada en un algoritmo genético multiobjetivo acoplada a un simulador de eventos discretos. La herramienta se llamó ONE (Optimization methodologies for Networke enterprises) .Su arquitectura es compuesta por cuatro módulos principales: Módulo de red, Módulo de optimización; Módulo de minería de datos y el módulo de simulación.

D. Dominio de los Informes,(dominios: Inteligencia, Clasificación Formulación del problema)

1. Simulation Optimization: Methods and applications
2. Simulation optimization research and development
3. Simulation optimization with qualitative variables and structural model changes: A genetic algorithm
4. Simulation Optimization Methodologies
5. A survey of simulation optimization techniques and procedures
6. Towards a framework for black box simulation optimization
7. Simulation Optimization
8. Integration of simulation and optimization for solving complex decision making problems
9. Simulation and optimization as effective DFSS tools

D1 Relación de los años de publicación de los estudios de tendencias

1. 1997
2. 1998
3. 1999
4. 1999
5. 2000
6. 2001
7. 2002
8. 2003
9. 2005

E. Aplicación a software comerciales*

1. Simulation modeling and optimization using Promodel
2. New advances for wedding optimization and simulation
3. Global search strategies for simulation optimization
4. The power of using automatic device optimization, based on iterative device-simulations, in design of high performance devices
1. Describe las características del paquete SimRunner
2. Introducción sobre simulación/Optimización y presenta el paquete Optquest
3. Describe la aplicación del algoritmo de optimización "swarm optimization" utilizado en el paquete Simul8
4. Aplicación de un proceso iterativo de simulación optimización al proyecto de equipos industriales con una demostración de la utilidad del método y una comparación entre diferentes topologías de equipos. Uso de MATLAB

A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II

Kalyanmoy Deb, *Associate Member, IEEE*, Amrit Pratap, Sameer Agarwal, and T. Meyarivan

Abstract—Multiobjective evolutionary algorithms (EAs) that use nondominated sorting and sharing have been criticized mainly for their: 1) $O(MN^2)$ computational complexity (where M is the number of objectives and N is the population size); 2) nonelitism approach; and 3) the need for specifying a sharing parameter. In this paper, we suggest a nondominated sorting-based multiobjective EA (MOEA), called nondominated sorting genetic algorithm II (NSGA-II), which alleviates all the above three difficulties. Specifically, a fast nondominated sorting approach with $O(MN^2)$ computational complexity is presented. Also, a selection operator is presented that creates a mating pool by combining the parent and offspring populations and selecting the best (with respect to fitness and spread) N solutions. Simulation results on difficult test problems show that the proposed NSGA-II, in most problems, is able to find much better spread of solutions and better convergence near the true Pareto-optimal front compared to Pareto-archived evolution strategy and strength-Pareto EA—two other elitist MOEAs that pay special attention to creating a diverse Pareto-optimal front. Moreover, we modify the definition of dominance in order to solve constrained multiobjective problems efficiently. Simulation results of the constrained NSGA-II on a number of test problems, including a five-objective seven-constraint nonlinear problem, are compared with another constrained multiobjective optimizer and much better performance of NSGA-II is observed.

Index Terms—Constraint handling, elitism, genetic algorithms, multicriterion decision making, multiobjective optimization, Pareto-optimal solutions.

I. INTRODUCTION

THE PRESENCE of multiple objectives in a problem, in principle, gives rise to a set of optimal solutions (largely known as Pareto-optimal solutions), instead of a single optimal solution. In the absence of any further information, one of these Pareto-optimal solutions cannot be said to be better than the other. This demands a user to find as many Pareto-optimal solutions as possible. Classical optimization methods (including the multicriterion decision-making methods) suggest converting the multiobjective optimization problem to a single-objective optimization problem by emphasizing one particular Pareto-optimal solution at a time. When such a method is to be used for finding multiple solutions, it has to be applied many times, hopefully finding a different solution at each simulation run.

Over the past decade, a number of multiobjective evolutionary algorithms (MOEAs) have been suggested [1], [7], [13],

[20], [26]. The primary reason for this is their ability to find multiple Pareto-optimal solutions in one single simulation run. Since evolutionary algorithms (EAs) work with a population of solutions, a simple EA can be extended to maintain a diverse set of solutions. With an emphasis for moving toward the true Pareto-optimal region, an EA can be used to find multiple Pareto-optimal solutions in one single simulation run.

The nondominated sorting genetic algorithm (NSGA) proposed in [20] was one of the first such EAs. Over the years, the main criticisms of the NSGA approach have been as follows.

- 1) *High computational complexity of nondominated sorting:* The currently-used nondominated sorting algorithm has a computational complexity of $O(MN^3)$ (where M is the number of objectives and N is the population size). This makes NSGA computationally expensive for large population sizes. This large complexity arises because of the complexity involved in the nondominated sorting procedure in every generation.
- 2) *Lack of elitism:* Recent results [25], [18] show that elitism can speed up the performance of the GA significantly, which also can help preventing the loss of good solutions once they are found.
- 3) *Need for specifying the sharing parameter σ_{share} :* Traditional mechanisms of ensuring diversity in a population so as to get a wide variety of equivalent solutions have relied mostly on the concept of sharing. The main problem with sharing is that it requires the specification of a sharing parameter (σ_{share}). Though there has been some work on dynamic sizing of the sharing parameter [10], a parameter-less diversity-preservation mechanism is desirable.

In this paper, we address all of these issues and propose an improved version of NSGA, which we call NSGA-II. From the simulation results on a number of difficult test problems, we find that NSGA-II outperforms two other contemporary MOEAs: Pareto-archived evolution strategy (PAES) [14] and strength-Pareto EA (SPEA) [24] in terms of finding a diverse set of solutions and in converging near the true Pareto-optimal set.

Constrained multiobjective optimization is important from the point of view of practical problem solving, but not much attention has been paid so far in this respect among the EA researchers. In this paper, we suggest a simple constraint-handling strategy with NSGA-II that suits well for any EA. On four problems chosen from the literature, NSGA-II has been compared with another recently suggested constraint-handling strategy. These results encourage the application of NSGA-II to more complex and real-world multiobjective optimization problems.

In the remainder of the paper, we briefly mention a number of existing elitist MOEAs in Section II. Thereafter, in Section III,

Manuscript received August 18, 2000; revised February 5, 2001 and September 7, 2001. The work of K. Deb was supported by the Ministry of Human Resources and Development, India, under the Research and Development Scheme.

The authors are with the Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology, Kanpur PIN 208 016, India (e-mail: deb@iitk.ac.in).

Publisher Item Identifier S 1089-778X/02/04101-2.

we describe the proposed NSGA-II algorithm in details. Section IV presents simulation results of NSGA-II and compares them with two other elitist MOEAs (PAES and SPEA). In Section V, we highlight the issue of parameter interactions, a matter that is important in evolutionary computation research. The next section extends NSGA-II for handling constraints and compares the results with another recently proposed constraint-handling method. Finally, we outline the conclusions of this paper.

II. ELITIST MULTIOBJECTIVE EVOLUTIONARY ALGORITHMS

During 1993–1995, a number of different EAs were suggested to solve multiobjective optimization problems. Of them, Fonseca and Fleming's MOGA [7], Srinivas and Deb's NSGA [20], and Horn *et al.*'s NPGA [13] enjoyed more attention. These algorithms demonstrated the necessary additional operators for converting a simple EA to a MOEA. Two common features on all three operators were the following: i) assigning fitness to population members based on nondominated sorting and ii) preserving diversity among solutions of the same nondominated front. Although they have been shown to find multiple nondominated solutions on many test problems and a number of engineering design problems, researchers realized the need of introducing more useful operators (which have been found useful in single-objective EA's) so as to solve multiobjective optimization problems better. Particularly, the interest has been to introduce elitism to enhance the convergence properties of a MOEA. Reference [25] showed that elitism helps in achieving better convergence in MOEAs. Among the existing elitist MOEAs, Zitzler and Thiele's SPEA [26], Knowles and Corne's Pareto-archived PAES [14], and Rudolph's elitist GA [18] are well studied. We describe these approaches in brief. For details, readers are encouraged to refer to the original studies.

Zitzler and Thiele [26] suggested an elitist multicriterion EA with the concept of nondomination in their SPEA. They suggested maintaining an external population at every generation storing all nondominated solutions discovered so far beginning from the initial population. This external population participates in all genetic operations. At each generation, a combined population with the external and the current population is first constructed. All nondominated solutions in the combined population are assigned a fitness based on the number of solutions they dominate and dominated solutions are assigned fitness worse than the worst fitness of any nondominated solution. This assignment of fitness makes sure that the search is directed toward the nondominated solutions. A deterministic clustering technique is used to ensure diversity among nondominated solutions. Although the implementation suggested in [26] is $O(MN^3)$, with proper bookkeeping the complexity of SPEA can be reduced to $O(MN^2)$.

Knowles and Corne [14] suggested a simple MOEA using a single-parent single-offspring EA similar to (1+1)-evolution strategy. Instead of using real parameters, binary strings were used and bitwise mutations were employed to create offsprings. In their PAES, with one parent and one offspring, the offspring is compared with respect to the parent. If the offspring dominates the parent, the offspring is accepted as the next parent and

the iteration continues. On the other hand, if the parent dominates the offspring, the offspring is discarded and a new mutated solution (a new offspring) is found. However, if the offspring and the parent do not dominate each other, the choice between the offspring and the parent is made by comparing them with an archive of best solutions found so far. The offspring is compared with the archive to check if it dominates any member of the archive. If it does, the offspring is accepted as the new parent and all the dominated solutions are eliminated from the archive. If the offspring does not dominate any member of the archive, both parent and offspring are checked for their *nearness* with the solutions of the archive. If the offspring resides in a least crowded region in the objective space among the members of the archive, it is accepted as a parent and a copy of added to the archive. Crowding is maintained by dividing the entire search space deterministically in d^n subspaces, where d is the depth parameter and n is the number of decision variables, and by updating the subspaces dynamically. Investigators have calculated the worst case complexity of PAES for N evaluations as $O(aMN)$, where a is the archive length. Since the archive size is usually chosen proportional to the population size N , the overall complexity of the algorithm is $O(MN^2)$.

Rudolph [18] suggested, but did not simulate, a simple elitist MOEA based on a systematic comparison of individuals from parent and offspring populations. The nondominated solutions of the offspring population are compared with that of parent solutions to form an overall nondominated set of solutions, which becomes the parent population of the next iteration. If the size of this set is not greater than the desired population size, other individuals from the offspring population are included. With this strategy, he proved the convergence of this algorithm to the Pareto-optimal front. Although this is an important achievement in its own right, the algorithm lacks motivation for the second task of maintaining diversity of Pareto-optimal solutions. An explicit diversity-preserving mechanism must be added to make it more practical. Since the determinism of the first nondominated front is $O(MN^2)$, the overall complexity of Rudolph's algorithm is also $O(MN^2)$.

In the following, we present the proposed nondominated sorting GA approach, which uses a fast nondominated sorting procedure, an elitist-preserving approach, and a parameterless niching operator.

III. ELITIST NONDOMINATED SORTING GENETIC ALGORITHM

A. Fast Nondominated Sorting Approach

For the sake of clarity, we first describe a naive and slow procedure of sorting a population into different nondomination levels. Thereafter, we describe a fast approach.

In a naive approach, in order to identify solutions of the first nondominated front in a population of size N , each solution can be compared with every other solution in the population to find if it is dominated. This requires $O(MN)$ comparisons for each solution, where M is the number of objectives. When this process is continued to find all members of the first nondominated level in the population, the total complexity is $O(MN^2)$. At this stage, all individuals in the first nondominated front are found. In order to find the individuals in the next nondominated

front, the solutions of the first front are discounted temporarily and the above procedure is repeated. In the worst case, the task of finding the second front also requires $O(MN^2)$ computations, particularly when $O(N)$ number of solutions belong to the second and higher nondominated levels. This argument is true for finding third and higher levels of nondomination. Thus, the worst case is when there are N fronts and there exists only one solution in each front. This requires an overall $O(MN^3)$ computations. Note that $O(N)$ storage is required for this procedure. In the following paragraph and equation shown at the bottom of the page, we describe a fast nondominated sorting approach which will require $O(MN^2)$ computations.

First, for each solution we calculate two entities: 1) domination count n_p , the number of solutions which dominate the solution p , and 2) S_p , a set of solutions that the solution p dominates. This requires $O(MN^2)$ comparisons.

All solutions in the first nondominated front will have their domination count as zero. Now, for each solution p with $n_p = 0$, we visit each member (q) of its set S_p and reduce its domination count by one. In doing so, if for any member q the domination count becomes zero, we put it in a separate list Q . These members belong to the second nondominated front. Now, the above procedure is continued with each member of Q and the third front is identified. This process continues until all fronts are identified.

For each solution p in the second or higher level of nondomination, the domination count n_p can be at most $N - 1$. Thus, each solution p will be visited at most $N - 1$ times before its domination count becomes zero. At this point, the solution is assigned a nondomination level and will never be visited again. Since there are at most $N - 1$ such solutions, the total com-

plexity is $O(N^2)$. Thus, the overall complexity of the procedure is $O(MN^2)$. Another way to calculate this complexity is to realize that the body of the first inner loop (for each $p \in \mathcal{F}_i$) is executed exactly N times as each individual can be the member of at most one front and the second inner loop (for each $q \in S_p$) can be executed at maximum $(N - 1)$ times for each individual [each individual dominates $(N - 1)$ individuals at maximum and each domination check requires at most M comparisons] results in the overall $O(MN^2)$ computations. It is important to note that although the time complexity has reduced to $O(MN^2)$, the storage requirement has increased to $O(N^2)$.

B. Diversity Preservation

We mentioned earlier that, along with convergence to the Pareto-optimal set, it is also desired that an EA maintains a good spread of solutions in the obtained set of solutions. The original NSGA used the well-known sharing function approach, which has been found to maintain sustainable diversity in a population with appropriate setting of its associated parameters. The sharing function method involves a sharing parameter σ_{share} , which sets the extent of sharing desired in a problem. This parameter is related to the distance metric chosen to calculate the proximity measure between two population members. The parameter σ_{share} denotes the largest value of that distance metric within which any two solutions share each other's fitness. This parameter is usually set by the user, although there exist some guidelines [4]. There are two difficulties with this sharing function approach.

- 1) The performance of the sharing function method in maintaining a spread of solutions depends largely on the chosen σ_{share} value.

```

fast-non-dominated-sort( $P$ )
for each  $p \in P$ 
   $S_p = \emptyset$ 
   $n_p = 0$ 
  for each  $q \in P$ 
    if ( $p \prec q$ ) then
       $S_p = S_p \cup \{q\}$ 
      Add  $q$  to the set of solutions dominated by  $p$ 
    else if ( $q \prec p$ ) then
      Increment the domination counter of  $p$ 
       $n_p = n_p + 1$ 
       $p$  belongs to the first front
  if  $n_p = 0$  then
     $p_{rank} = 1$ 
     $\mathcal{F}_1 = \mathcal{F}_1 \cup \{p\}$ 
  Initialize the front counter
 $i = 1$ 
while  $\mathcal{F}_i \neq \emptyset$ 
   $Q = \emptyset$ 
  Used to store the members of the next front
  for each  $p \in \mathcal{F}_i$ 
    for each  $q \in S_p$ 
       $n_q = n_q - 1$ 
      if  $n_q = 0$  then
         $q$  belongs to the next front
         $q_{rank} = i + 1$ 
         $Q = Q \cup \{q\}$ 
   $i = i + 1$ 
   $\mathcal{F}_i = Q$ 

```

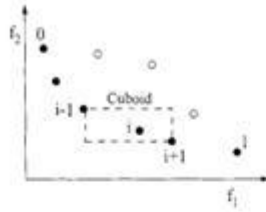


Fig. 1. Crowding-distance calculation. Points marked in filled circles are solutions of the same nondominated front.

- 2) Since each solution must be compared with all other solutions in the population, the overall complexity of the sharing function approach is $O(N^2)$.

In the proposed NSGA-II, we replace the sharing function approach with a crowded-comparison approach that eliminates both the above difficulties to some extent. The new approach does not require *any* user-defined parameter for maintaining diversity among population members. Also, the suggested approach has a better computational complexity. To describe this approach, we first define a density-estimation metric and then present the crowded-comparison operator.

1) *Density Estimation*: To get an estimate of the density of solutions surrounding a particular solution in the population, we calculate the average distance of two points on either side of this point along each of the objectives. This quantity $i_{distance}$ serves as an estimate of the perimeter of the cuboid formed by using the nearest neighbors as the vertices (call this the *crowding distance*). In Fig. 1, the crowding distance of the i th solution in its front (marked with solid circles) is the average side length of the cuboid (shown with a dashed box).

The crowding-distance computation requires sorting the population according to each objective function value in ascending order of magnitude. Thereafter, for each objective function, the boundary solutions (solutions with smallest and largest function values) are assigned an infinite distance value. All other intermediate solutions are assigned a distance value equal to the absolute normalized difference in the function values of two adjacent solutions. This calculation is continued with other objective functions. The overall crowding-distance value is calculated as the sum of individual distance values corresponding to each objective. Each objective function is normalized before calculating the crowding distance. The algorithm as shown at the bottom of the page outlines the crowding-distance computation procedure of all solutions in an nondominated set \mathcal{I} .

Here, $\mathcal{I}[i].m$ refers to the m th objective function value of the i th individual in the set \mathcal{I} and the parameters f_m^{max} and f_m^{min} are the maximum and minimum values of the m th objective function. The complexity of this procedure is governed by the sorting algorithm. Since M independent sortings of at most N solutions (when all population members are in one front \mathcal{I}) are involved, the above algorithm has $O(MN \log N)$ computational complexity.

After all population members in the set \mathcal{I} are assigned a distance metric, we can compare two solutions for their extent of proximity with other solutions. A solution with a smaller value of this distance measure is, in some sense, more crowded by other solutions. This is exactly what we compare in the proposed crowded-comparison operator, described below. Although Fig. 1 illustrates the crowding-distance computation for two objectives, the procedure is applicable to more than two objectives as well.

2) *Crowded-Comparison Operator*: The crowded-comparison operator ($<_n$) guides the selection process at the various stages of the algorithm toward a uniformly spread-out Pareto-optimal front. Assume that every individual i in the population has two attributes:

- 1) nondomination rank (i_{rank});
- 2) crowding distance ($i_{distance}$).

We now define a partial order $<_n$ as

$$i <_n j \quad \text{if } (i_{rank} < j_{rank}) \\ \text{or } ((i_{rank} = j_{rank}) \\ \text{and } (i_{distance} > j_{distance}))$$

That is, between two solutions with differing nondomination ranks, we prefer the solution with the lower (better) rank. Otherwise, if both solutions belong to the same front, then we prefer the solution that is located in a lesser crowded region.

With these three new innovations—a fast nondominated sorting procedure, a fast crowded distance estimation procedure, and a simple crowded comparison operator, we are now ready to describe the NSGA-II algorithm.

C. Main Loop

Initially, a random parent population P_0 is created. The population is sorted based on the nondomination. Each solution is assigned a fitness (or rank) equal to its nondomination level (1 is the best level, 2 is the next-best level, and so on). Thus, minimization of fitness is assumed. At first, the usual binary tournament selection, recombination, and mutation operators are used to create a offspring population Q_0 of size N . Since elitism

```
crowding-distance-assignment( $\mathcal{I}$ )
```

```
 $l = |\mathcal{I}|$ 
```

```
for each  $i$ , set  $\mathcal{I}[i].distance = 0$ 
```

```
for each objective  $m$ 
```

```
   $\mathcal{I} = \text{sort}(\mathcal{I}, m)$ 
```

```
   $\mathcal{I}[1].distance = \mathcal{I}[l].distance = \infty$ 
```

```
  for  $i = 2$  to  $(l - 1)$ 
```

```
     $\mathcal{I}[i].distance = \mathcal{I}[i].distance + (\mathcal{I}[i + 1].m - \mathcal{I}[i - 1].m) / (f_m^{max} - f_m^{min})$ 
```

```
number of solutions in  $\mathcal{I}$ 
```

```
initialize distance
```

```
sort using each objective value
```

```
so that boundary points are always selected
```

```
for all other points
```

is introduced by comparing current population with previously found best nondominated solutions, the procedure is different after the initial generation. We first describe the t th generation of the proposed algorithm as shown at the bottom of the page.

The step-by-step procedure shows that NSGA-II algorithm is simple and straightforward. First, a combined population $R_t = P_t \cup Q_t$ is formed. The population R_t is of size $2N$. Then, the population R_t is sorted according to nondomination. Since all previous and current population members are included in R_t , elitism is ensured. Now, solutions belonging to the best nondominated set \mathcal{F}_1 are of best solutions in the combined population and must be emphasized more than any other solution in the combined population. If the size of \mathcal{F}_1 is smaller than N , we definitely choose all members of the set \mathcal{F}_1 for the new population P_{t+1} . The remaining members of the population P_{t+1} are chosen from subsequent nondominated fronts in the order of their ranking. Thus, solutions from the set \mathcal{F}_2 are chosen next, followed by solutions from the set \mathcal{F}_3 , and so on. This procedure is continued until no more sets can be accommodated. Say that the set \mathcal{F}_i is the last nondominated set beyond which no other set can be accommodated. In general, the count of solutions in all sets from \mathcal{F}_1 to \mathcal{F}_i would be larger than the population size. To choose exactly N population members, we sort the solutions of the last front \mathcal{F}_i using the crowded-comparison operator \prec_n in descending order and choose the best solutions needed to fill all population slots. The NSGA-II procedure is also shown in Fig. 2. The new population P_{t+1} of size N is now used for selection, crossover, and mutation to create a new population Q_{t+1} of size N . It is important to note that we use a binary tournament selection operator but the selection criterion is now based on the crowded-comparison operator \prec_n . Since this operator requires both the rank and crowded distance of each solution in the population, we calculate these quantities while forming the population P_{t+1} , as shown in the above algorithm.

Consider the complexity of one iteration of the entire algorithm. The basic operations and their worst-case complexities are as follows:

- 1) nondominated sorting is $O(M(2N)^2)$;
- 2) crowding-distance assignment is $O(M(2N) \log(2N))$;
- 3) sorting on \prec_n is $O(2N \log(2N))$.

The overall complexity of the algorithm is $O(MN^2)$, which is governed by the nondominated sorting part of the algorithm. If

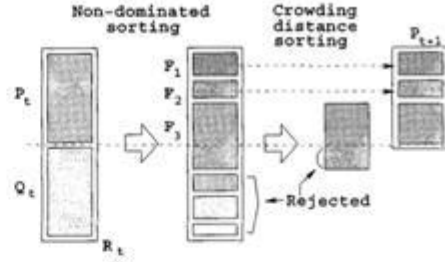


Fig. 2. NSGA-II procedure.

performed carefully, the complete population of size $2N$ need not be sorted according to nondomination. As soon as the sorting procedure has found enough number of fronts to have N members in P_{t+1} , there is no reason to continue with the sorting procedure.

The diversity among nondominated solutions is introduced by using the crowding comparison procedure, which is used in the tournament selection and during the population reduction phase. Since solutions compete with their crowding-distance (a measure of density of solutions in the neighborhood), no extra niching parameter (such as σ_{share} needed in the NSGA) is required. Although the crowding distance is calculated in the objective function space, it can also be implemented in the parameter space, if so desired [3]. However, in all simulations performed in this study, we have used the objective-function space niching.

IV. SIMULATION RESULTS

In this section, we first describe the test problems used to compare the performance of NSGA-II with PAES and SPEA. For PAES and SPEA, we have identical parameter settings as suggested in the original studies. For NSGA-II, we have chosen a reasonable set of values and have not made any effort in finding the best parameter setting. We leave this task for a future study.

<pre> $R_t = P_t \cup Q_t$ $\mathcal{F} = \text{fast-non-dominated-sort}(R_t)$ $P_{t+1} = \emptyset$ and $i = 1$ until $P_{t+1} + \mathcal{F}_i \leq N$ crowding-distance-assignment(\mathcal{F}_i) $P_{t+1} = P_{t+1} \cup \mathcal{F}_i$ $i = i + 1$ Sort(\mathcal{F}_i, \prec_n) $P_{t+1} = P_{t+1} \cup \mathcal{F}_i[1 : (N - P_{t+1})]$ $Q_{t+1} = \text{make-new-pop}(P_{t+1})$ $t = t + 1$ </pre>	<pre> combine parent and offspring population $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \dots)$, all nondominated fronts of R_t until the parent population is filled calculate crowding-distance in \mathcal{F}_i include ith nondominated front in the parent pop check the next front for inclusion sort in descending order using \prec_n choose the first $(N - P_{t+1})$ elements of \mathcal{F}_i use selection, crossover and mutation to create a new population Q_{t+1} increment the generation counter </pre>
--	--

TABLE I
TEST PROBLEMS USED IN THIS STUDY

Problem	n	Variable bounds	Objective functions	Optimal solutions	Comments
SCH	1	$[-10^3, 10^3]$	$f_1(x) = x^2$ $f_2(x) = (x-2)^2$	$x \in [0, 2]$	convex
FON	3	$[-4, 4]$	$f_1(\mathbf{x}) = 1 - \exp\left(-\sum_{i=1}^3 \left(x_i - \frac{1}{\sqrt{3}}\right)^2\right)$ $f_2(\mathbf{x}) = 1 - \exp\left(-\sum_{i=1}^3 \left(x_i + \frac{1}{\sqrt{3}}\right)^2\right)$	$x_1 = x_2 = x_3$ $\in [-1/\sqrt{3}, 1/\sqrt{3}]$	nonconvex
POL	2	$[-\pi, \pi]$	$f_1(\mathbf{x}) = [1 + (A_1 - B_1)^2 + (A_2 - B_2)^2]$ $f_2(\mathbf{x}) = [(x_1 + 3)^2 + (x_2 + 1)^2]$ $A_1 = 0.5 \sin 1 - 2 \cos 1 + \sin 2 - 1.5 \cos 2$ $A_2 = 1.5 \sin 1 - \cos 1 + 2 \sin 2 - 0.5 \cos 2$ $B_1 = 0.5 \sin x_1 - 2 \cos x_1 + \sin x_2 - 1.5 \cos x_2$ $B_2 = 1.5 \sin x_1 - \cos x_1 + 2 \sin x_2 - 0.5 \cos x_2$	(refer [1])	nonconvex, disconnected
KUR	3	$[-5, 5]$	$f_1(\mathbf{x}) = \sum_{i=1}^{n-1} \left(-10 \exp\left(-0.2 \sqrt{x_i^2 + x_{i+1}^2}\right)\right)$ $f_2(\mathbf{x}) = \sum_{i=1}^n (x_i ^{0.8} + 5 \sin x_i^1)$	(refer [1])	nonconvex
ZDT1	30	$[0, 1]$	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x}) \left[1 - \sqrt{x_1/g(\mathbf{x})}\right]$ $g(\mathbf{x}) = 1 + 9 \left(\sum_{i=2}^n x_i\right) / (n-1)$	$x_1 \in [0, 1]$ $x_i = 0,$ $i = 2, \dots, n$	convex
ZDT2	30	$[0, 1]$	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x}) \left[1 - (x_1/g(\mathbf{x}))^2\right]$ $g(\mathbf{x}) = 1 + 9 \left(\sum_{i=2}^n x_i\right) / (n-1)$	$x_1 \in [0, 1]$ $x_i = 0,$ $i = 2, \dots, n$	nonconvex
ZDT3	30	$[0, 1]$	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x}) \left[1 - \sqrt{x_1/g(\mathbf{x})} - \frac{n}{\pi} \sin(10\pi x_1)\right]$ $g(\mathbf{x}) = 1 + 9 \left(\sum_{i=2}^n x_i\right) / (n-1)$	$x_1 \in [0, 1]$ $x_i = 0,$ $i = 2, \dots, n$	convex, disconnected
ZDT4	10	$x_1 \in [0, 1]$ $x_i \in [-5, 5],$ $i = 2, \dots, n$	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x}) \left[1 - \sqrt{x_1/g(\mathbf{x})}\right]$ $g(\mathbf{x}) = 1 + 10(n-1) + \sum_{i=2}^n [x_i^2 - 10 \cos(4\pi x_i)]$	$x_1 \in [0, 1]$ $x_i = 0,$ $i = 2, \dots, n$	nonconvex
ZDT6	10	$[0, 1]$	$f_1(\mathbf{x}) = 1 - \exp(-4x_1) \sin^6(6\pi x_1)$ $f_2(\mathbf{x}) = g(\mathbf{x}) \left[1 - (f_1(\mathbf{x})/g(\mathbf{x}))^2\right]$ $g(\mathbf{x}) = 1 + 9 \left(\sum_{i=2}^n x_i\right) / (n-1)^{0.25}$	$x_1 \in [0, 1]$ $x_i = 0,$ $i = 2, \dots, n$	nonconvex, nonuniformly spaced

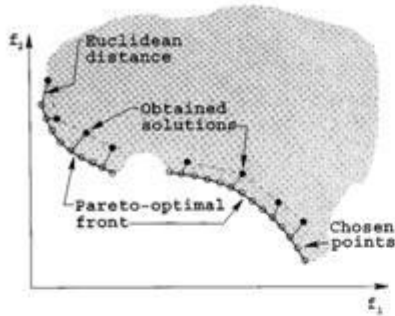
All objective functions are to be minimized.

A. Test Problems

We first describe the test problems used to compare different MOEAs. Test problems are chosen from a number of significant past studies in this area. Veldhuizen [22] cited a number of test problems that have been used in the past. Of them, we choose four problems: Schaffer's study (SCH) [19], Fonseca and Fleming's study (FON) [10], Poloni's study (POL) [16], and Kursawe's study (KUR) [15]. In 1999, the first author suggested a systematic way of developing test problems for multiobjective optimization [3]. Zitzler et al. [25] followed those guidelines and suggested six test problems. We choose five of those six problems here and call them ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6. All problems have two objective functions. None of these problems have any constraint. We describe these problems in Table I. The table also shows the number of variables, their bounds, the Pareto-optimal solutions, and the nature of the Pareto-optimal front for each problem.

All approaches are run for a maximum of 25 000 function evaluations. We use the single-point crossover and bitwise

mutation for binary-coded GAs and the simulated binary crossover (SBX) operator and polynomial mutation [6] for real-coded GAs. The crossover probability of $p_c = 0.9$ and a mutation probability of $p_m = 1/n$ or $1/\ell$ (where n is the number of decision variables for real-coded GAs and ℓ is the string length for binary-coded GAs) are used. For real-coded NSGA-II, we use distribution indexes [6] for crossover and mutation operators as $\eta_c = 20$ and $\eta_m = 20$, respectively. The population obtained at the end of 250 generations (the population after elite-preserving operator is applied) is used to calculate a couple of performance metrics, which we discuss in the next section. For PAES, we use a depth value d equal to four and an archive size a of 100. We use all population members of the archive obtained at the end of 25 000 iterations to calculate the performance metrics. For SPEA, we use a population of size 80 and an external population of size 20 (this 4:1 ratio is suggested by the developers of SPEA to maintain an adequate selection pressure for the elite solutions), so that overall population size becomes 100. SPEA is also run until 25 000 function evaluations are done. For SPEA, we use the

Fig. 3. Distance metric Υ .

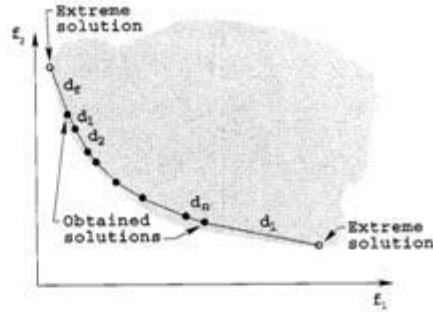
nondominated solutions of the combined GA and external populations at the final generation to calculate the performance metrics used in this study. For PAES, SPEA, and binary-coded NSGA-II, we have used 30 bits to code each decision variable.

B. Performance Measures

Unlike in single-objective optimization, there are two goals in a multiobjective optimization: 1) convergence to the Pareto-optimal set and 2) maintenance of diversity in solutions of the Pareto-optimal set. These two tasks cannot be measured adequately with one performance metric. Many performance metrics have been suggested [11], [8], [24]. Here, we define two performance metrics that are more direct in evaluating each of the above two goals in a solution set obtained by a multiobjective optimization algorithm.

The first metric Υ measures the extent of convergence to a known set of Pareto-optimal solutions. Since multiobjective algorithms would be tested on problems having a known set of Pareto-optimal solutions, the calculation of this metric is possible. We realize, however, that such a metric cannot be used for any arbitrary problem. First, we find a set of $H = 500$ uniformly spaced solutions from the true Pareto-optimal front in the objective space. For each solution obtained with an algorithm, we compute the minimum Euclidean distance of it from H chosen solutions on the Pareto-optimal front. The average of these distances is used as the first metric Υ (the convergence metric). Fig. 3 shows the calculation procedure of this metric. The shaded region is the feasible search region and the solid curved lines specify the Pareto-optimal solutions. Solutions with open circles are H chosen solutions on the Pareto-optimal front for the calculation of the convergence metric and solutions marked with dark circles are solutions obtained by an algorithm. The smaller the value of this metric, the better the convergence toward the Pareto-optimal front. When all obtained solutions lie exactly on H chosen solutions, this metric takes a value of zero. In all simulations performed here, we present the average $\bar{\Upsilon}$ and variance σ_{Υ} of this metric calculated for solution sets obtained in multiple runs.

Even when all solutions converge to the Pareto-optimal front, the above convergence metric does not have a value of zero. The metric will yield zero only when each obtained solution lies exactly on each of the chosen solutions. Although this metric alone

Fig. 4. Diversity metric Δ .

can provide some information about the spread in obtained solutions, we define a different metric to measure the spread in solutions obtained by an algorithm directly. The second metric Δ measures the extent of spread achieved among the obtained solutions. Here, we are interested in getting a set of solutions that spans the entire Pareto-optimal region. We calculate the Euclidean distance d_i between consecutive solutions in the obtained nondominated set of solutions. We calculate the average \bar{d} of these distances. Thereafter, from the obtained set of nondominated solutions, we first calculate the *extreme* solutions (in the objective space) by fitting a curve parallel to that of the true Pareto-optimal front. Then, we use the following metric to calculate the nonuniformity in the distribution:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_l + (N-1)\bar{d}} \quad (1)$$

Here, the parameters d_f and d_l are the Euclidean distances between the extreme solutions and the boundary solutions of the obtained nondominated set, as depicted in Fig. 4. The figure illustrates all distances mentioned in the above equation. The parameter \bar{d} is the average of all distances d_i , $i = 1, 2, \dots, (N-1)$, assuming that there are N solutions on the best nondominated front. With N solutions, there are $(N-1)$ consecutive distances. The denominator is the value of the numerator for the case when all N solutions lie on one solution. It is interesting to note that this is not the worst case spread of solutions possible. We can have a scenario in which there is a large variance in d_i . In such scenarios, the metric may be greater than one. Thus, the maximum value of the above metric can be greater than one. However, a good distribution would make all distances d_i equal to \bar{d} and would make $d_f = d_l = 0$ (with existence of extreme solutions in the nondominated set). Thus, for the most widely and uniformly spread-out set of nondominated solutions, the numerator of Δ would be zero, making the metric to take a value zero. For any other distribution, the value of the metric would be greater than zero. For two distributions having identical values of d_f and d_l , the metric Δ takes a higher value with worse distributions of solutions within the extreme solutions. Note that the above diversity metric can be used on any nondominated set of solutions, including one that is not the Pareto-optimal set. Using

TABLE II
MEAN (FIRST ROWS) AND VARIANCE (SECOND ROWS) OF THE CONVERGENCE METRIC Υ

Algorithm	SCH	FON	POL	KUR	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
NSGA-II	0.003391	0.001931	0.015553	0.028964	0.033482	0.072391	0.114500	0.513053	0.296564
Real-coded	0	0	0.000001	0.000018	0.004750	0.031689	0.007940	0.118460	0.013135
NSGA-II	0.002833	0.002571	0.017029	0.028951	0.000894	0.000824	0.043411	3.227636	7.806798
Binary-coded	0.000001	0	0.000003	0.000016	0	0	0.000042	7.30763	0.001667
SPEA	0.003403	0.125692	0.037812	0.045617	0.001799	0.001339	0.047517	7.340299	0.221138
	0	0.000038	0.000088	0.00005	0.000001	0	0.000047	6.572516	0.000449
PAES	0.001313	0.151263	0.030864	0.057323	0.082085	0.126276	0.023872	0.854816	0.085469
	0.000003	0.000905	0.000431	0.011989	0.008679	0.036877	0.00001	0.527238	0.006664

TABLE III
MEAN (FIRST ROWS) AND VARIANCE (SECOND ROWS) OF THE DIVERSITY METRIC Δ

Algorithm	SCH	FON	POL	KUR	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
NSGA2R	0.477899	0.378065	0.452150	0.411477	0.390307	0.430776	0.738540	0.702612	0.668025
Real-coded	0.003471	0.000639	0.002868	0.000992	0.001876	0.004721	0.019706	0.064648	0.009923
NSGA-II	0.449265	0.395131	0.503721	0.442195	0.463292	0.435112	0.575606	0.479475	0.644477
Binary-coded	0.002062	0.001314	0.004656	0.001498	0.041622	0.024607	0.005078	0.009841	0.035042
SPEA	1.021110	0.792352	0.972783	0.852990	0.784525	0.755148	0.672938	0.798463	0.849389
	0.004372	0.005546	0.008475	0.002619	0.004440	0.004521	0.003587	0.014616	0.002713
PAES	1.063288	1.162528	1.020007	1.079838	1.229794	1.165942	0.789920	0.870458	1.153052
	0.002868	0.008945	0	0.013772	0.004839	0.007682	0.001653	0.101399	0.003916

a triangularization technique or a Voronoi diagram approach [1] to calculate d_i , the above procedure can be extended to estimate the spread of solutions in higher dimensions.

C. Discussion of the Results

Table II shows the mean and variance of the convergence metric Υ obtained using four algorithms NSGA-II (real-coded), NSGA-II (binary-coded), SPEA, and PAES.

NSGA-II (real coded or binary coded) is able to converge better in all problems except in ZDT3 and ZDT6, where PAES found better convergence. In all cases with NSGA-II, the variance in ten runs is also small, except in ZDT4 with NSGA-II (binary coded). The fixed archive strategy of PAES allows better convergence to be achieved in two out of nine problems.

Table III shows the mean and variance of the diversity metric Δ obtained using all three algorithms.

NSGA-II (real or binary coded) performs the best in all nine test problems. The worst performance is observed with PAES. For illustration, we show one of the ten runs of PAES with an arbitrary run of NSGA-II (real-coded) on problem SCH in Fig. 5.

On most problems, real-coded NSGA-II is able to find a better spread of solutions than any other algorithm, including binary-coded NSGA-II.

In order to demonstrate the working of these algorithms, we also show typical simulation results of PAES, SPEA, and NSGA-II on the test problems KUR, ZDT2, ZDT4, and ZDT6. The problem KUR has three discontinuous regions in the Pareto-optimal front. Fig. 6 shows all nondominated solutions obtained after 250 generations with NSGA-II (real-coded). The Pareto-optimal region is also shown in the figure. This figure demonstrates the abilities of NSGA-II in converging to the true front and in finding diverse solutions in the front. Fig. 7 shows the obtained nondominated solutions with SPEA, which is the next-best algorithm for this problem (refer to Tables II and III).

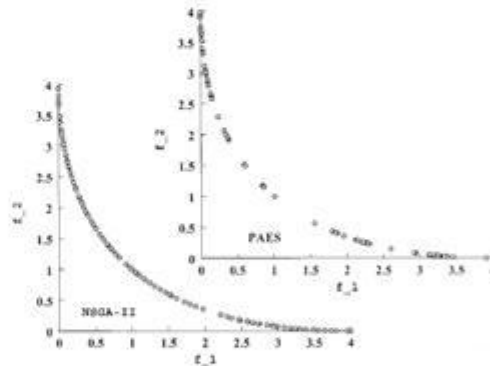


Fig. 5. NSGA-II finds better spread of solutions than PAES on SCH.

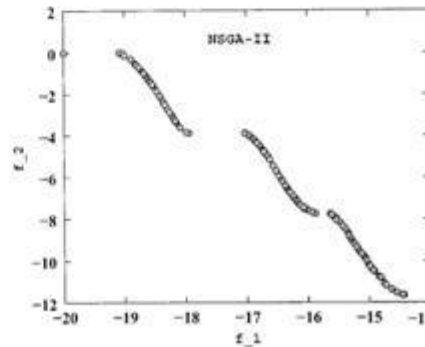


Fig. 6. Nondominated solutions with NSGA-II (real-coded) on KUR.

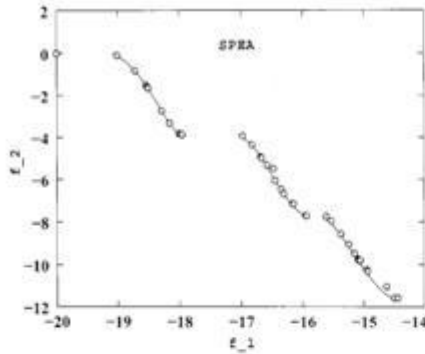


Fig. 7. Nondominated solutions with SPEA on KUR.

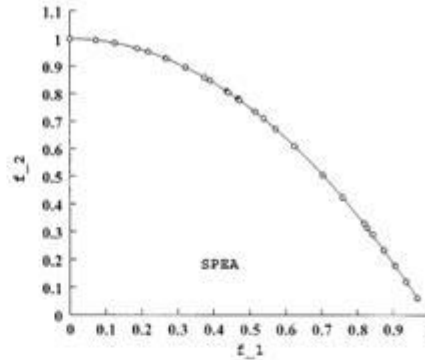


Fig. 9. Nondominated solutions with SPEA on ZDT2.

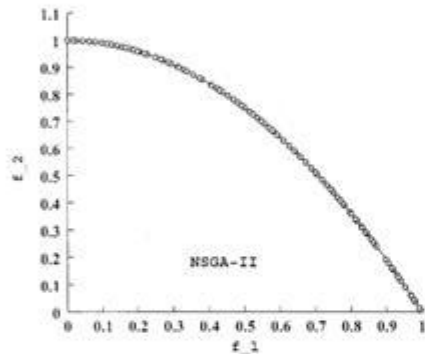


Fig. 8. Nondominated solutions with NSGA-II (binary-coded) on ZDT2.

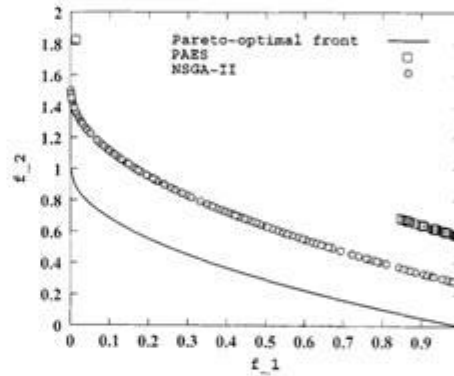


Fig. 10. NSGA-II finds better convergence and spread of solutions than PAES on ZDT4.

In both aspects of convergence and distribution of solutions, NSGA-II performed better than SPEA in this problem. Since SPEA could not maintain enough nondominated solutions in the final GA population, the overall number of nondominated solutions is much less compared to that obtained in the final population of NSGA-II.

Next, we show the nondominated solutions on the problem ZDT2 in Figs. 8 and 9. This problem has a nonconvex Pareto-optimal front. We show the performance of binary-coded NSGA-II and SPEA on this function. Although the convergence is not a difficulty here with both of these algorithms, both real- and binary-coded NSGA-II have found a better spread and more solutions in the entire Pareto-optimal region than SPEA (the next-best algorithm observed for this problem).

The problem ZDT4 has 21^9 or $7.94(10^{11})$ different local Pareto-optimal fronts in the search space, of which only one corresponds to the global Pareto-optimal front. The Euclidean distance in the decision space between solutions of two consecutive local Pareto-optimal sets is 0.25. Fig. 10 shows that both real-coded NSGA-II and PAES get stuck at different local Pareto-optimal sets, but the convergence and ability to find a diverse set of solutions are definitely better with NSGA-II. Binary-coded GAs have difficulties in converging

near the global Pareto-optimal front, a matter that is also been observed in previous single-objective studies [5]. On a similar ten-variable Rastrigin's function [the function $g(x)$ here], that study clearly showed that a population of size of about at least 500 is needed for single-objective binary-coded GAs (with tournament selection, single-point crossover and bitwise mutation) to find the global optimum solution in more than 50% of the simulation runs. Since we have used a population of size 100, it is not expected that a multiobjective GA would find the global Pareto-optimal solution, but NSGA-II is able to find a good spread of solutions even at a local Pareto-optimal front. Since SPEA converges poorly on this problem (see Table II), we do not show SPEA results on this figure.

Finally, Fig. 11 shows that SPEA finds a better converged set of nondominated solutions in ZDT6 compared to any other algorithm. However, the distribution in solutions is better with real-coded NSGA-II.

D. Different Parameter Settings

In this study, we do not make any serious attempt to find the best parameter setting for NSGA-II. But in this section, we per-

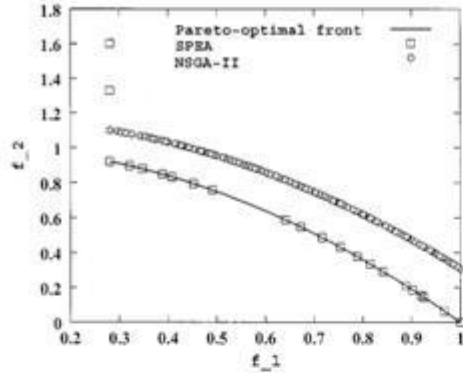


Fig. 11. Real-coded NSGA-II finds better spread of solutions than SPEA on ZDT6, but SPEA has a better convergence.

TABLE IV
MEAN AND VARIANCE OF THE CONVERGENCE AND DIVERSITY METRICS UP TO 500 GENERATIONS

Convergence metric, \bar{Y}					
	POL	KUR	ZDT3	ZDT4	ZDT6
Mean	0.015882	0.026544	0.018510	0.090692	0.276609
Variance	0.000001	0.000017	0.000227	0.053460	0.015843
Diversity metric, $\bar{\Delta}$					
	POL	KUR	ZDT3	ZDT4	ZDT6
Mean	0.467022	0.418889	0.688218	0.440022	0.655896
Variance	0.002186	0.000530	0.000610	0.026729	0.003302

form additional experiments to show the effect of a couple of different parameter settings on the performance of NSGA-II.

First, we keep all other parameters as before, but increase the number of maximum generations to 500 (instead of 250 used before). Table IV shows the convergence and diversity metrics for problems POL, KUR, ZDT3, ZDT4, and ZDT6. Now, we achieve a convergence very close to the true Pareto-optimal front and with a much better distribution. The table shows that in all these difficult problems, the real-coded NSGA-II has converged very close to the true optimal front, except in ZDT6, which probably requires a different parameter setting with NSGA-II. Particularly, the results on ZDT3 and ZDT4 improve with generation number.

The problem ZDT4 has a number of local Pareto-optimal fronts, each corresponding to particular value of $g(\mathbf{x})$. A large change in the decision vector is needed to get out of a local optimum. Unless mutation or crossover operators are capable of creating solutions in the basin of another better attractor, the improvement in the convergence toward the true Pareto-optimal front is not possible. We use NSGA-II (real-coded) with a smaller distribution index $\eta_m = 10$ for mutation, which has an effect of creating solutions with more spread than before. Rest of the parameter settings are identical as before. The convergence metric \bar{Y} and diversity measure $\bar{\Delta}$ on problem ZDT4 at the end of 250 generations are as follows:

$$\begin{aligned} \bar{Y} &= 0.029544 & \sigma_Y^2 &= 0.002145 \\ \bar{\Delta} &= 0.498409 & \sigma_{\Delta}^2 &= 0.003852. \end{aligned}$$

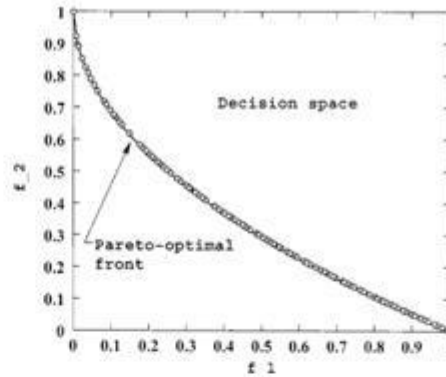


Fig. 12. Obtained nondominated solutions with NSGA-II on problem ZDT4.

These results are much better than PAES and SPEA, as shown in Table II. To demonstrate the convergence and spread of solutions, we plot the nondominated solutions of one of the runs after 250 generations in Fig. 12. The figure shows that NSGA-II is able to find solutions on the true Pareto-optimal front with $g(\mathbf{x}) = 1.0$.

V. ROTATED PROBLEMS

It has been discussed in an earlier study [3] that interactions among decision variables can introduce another level of difficulty to any multiobjective optimization algorithm including EAs. In this section, we create one such problem and investigate the working of previously three MOEAs on the following epistatic problem:

$$\begin{aligned} &\text{minimize } f_1(\mathbf{y}) = y_1 \\ &\text{minimize } f_2(\mathbf{y}) = g(\mathbf{y}) \exp(-y_1/g(\mathbf{y})) \\ &\text{where } g(\mathbf{y}) = 1 + 10(n-1) \\ &\quad + \sum_{i=2}^n [y_i^2 - 10 \cos(4\pi y_i)] \\ &\text{and } \mathbf{y} = \mathcal{R}\mathbf{x} \\ &\quad -0.3 \leq x_i \leq 0.3, \quad \text{for } i = 1, 2, \dots, n. \end{aligned} \quad (2)$$

An EA works with the decision variable vector \mathbf{x} , but the above objective functions are defined in terms of the variable vector \mathbf{y} , which is calculated by transforming the decision variable vector \mathbf{x} by a fixed rotation matrix \mathcal{R} . This way, the objective functions are functions of a linear combination of decision variables. In order to maintain a spread of solutions over the Pareto-optimal region or even converge to any particular solution requires an EA to update all decision variables in a particular fashion. With a generic search operator, such as the variablewise SBX operator used here, this becomes a difficult task for an EA. However, here, we are interested in evaluating the overall behavior of three elitist MOEAs.

We use a population size of 100 and run each algorithm until 500 generations. For SBX, we use $\eta_c = 10$ and we use $\eta_m = 50$ for mutation. To restrict the Pareto-optimal solutions to lie

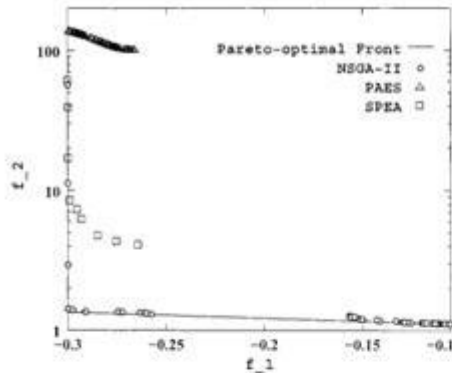


Fig. 13. Obtained nondominated solutions with NSGA-II, PAES, and SPEA on the rotated problem.

within the prescribed variable bounds, we discourage solutions with $|f_1| > 0.3$ by adding a fixed large penalty to both objectives. Fig. 13 shows the obtained solutions at the end of 500 generations using NSGA-II, PAES, and SPEA. It is observed that NSGA-II solutions are closer to the true front compared to solutions obtained by PAES and SPEA. The correlated parameter updates needed to progress toward the Pareto-optimal front makes this kind of problems difficult to solve. NSGA-II's elite-preserving operator along with the real-coded crossover and mutation operators is able to find some solutions close to the Pareto-optimal front [with $g(y) = 1$ resulting $f_2 = \exp(-f_1)$]. This example problem demonstrates that one of the known difficulties (the linkage problem [11], [12]) of single-objective optimization algorithm can also cause difficulties in a multiobjective problem. However, more systematic studies are needed to amply address the linkage issue in multiobjective optimization.

VI. CONSTRAINT HANDLING

In the past, the first author and his students implemented a penalty-parameterless constraint-handling approach for single-objective optimization. Those studies [2], [6] have shown how a tournament selection based algorithm can be used to handle constraints in a population approach much better than a number of other existing constraint-handling approaches. A similar approach can be introduced with the above NSGA-II for solving constrained multiobjective optimization problems.

A. Proposed Constraint-Handling Approach (Constrained NSGA-II)

This constraint-handling method uses the binary tournament selection, where two solutions are picked from the population and the better solution is chosen. In the presence of constraints, each solution can be either feasible or infeasible. Thus, there may be at most three situations: 1) both solutions are feasible; 2) one is feasible and other is not; and 3) both are infeasible.

For single objective optimization, we used a simple rule for each case.

- Case 1) Choose the solution with better objective function value.
- Case 2) Choose the feasible solution.
- Case 3) Choose the solution with smaller overall constraint violation.

Since in no case constraints and objective function values are compared with each other, there is no need of having any penalty parameter, a matter that makes the proposed constraint-handling approach useful and attractive.

In the context of multiobjective optimization, the latter two cases can be used as they are and the first case can be resolved by using the crowded-comparison operator as before. To maintain the modularity in the procedures of NSGA-II, we simply modify the definition of *domination* between two solutions i and j .

Definition 1: A solution i is said to constrained-dominate a solution j , if any of the following conditions is true.

- 1) Solution i is feasible and solution j is not.
- 2) Solutions i and j are both infeasible, but solution i has a smaller overall constraint violation.
- 3) Solutions i and j are feasible and solution i dominates solution j .

The effect of using this constrained-domination principle is that any feasible solution has a better nondomination rank than any infeasible solution. All feasible solutions are ranked according to their nondomination level based on the objective function values. However, among two infeasible solutions, the solution with a smaller constraint violation has a better rank. Moreover, this modification in the nondomination principle does not change the computational complexity of NSGA-II. The rest of the NSGA-II procedure as described earlier can be used as usual.

The above constrained-domination definition is similar to that suggested by Fonseca and Fleming [9]. The only difference is in the way domination is defined for the infeasible solutions. In the above definition, an infeasible solution having a larger overall constraint-violation are classified as members of a larger nondomination level. On the other hand, in [9], infeasible solutions violating different constraints are classified as members of the same nondominated front. Thus, one infeasible solution violating a constraint marginally will be placed in the same nondominated level with another solution violating a different constraint to a large extent. This may cause an algorithm to wander in the infeasible search region for more generations before reaching the feasible region through constraint boundaries. Moreover, since Fonseca-Fleming's approach requires domination checks with the constraint-violation values, the proposed approach of this paper is computationally less expensive and is simpler.

B. Ray-Tai-Seow's Constraint-Handling Approach

Ray *et al.* [17] suggested a more elaborate constraint-handling technique, where constraint violations of all constraints are not simply summed together. Instead, a nondomination check of constraint violations is also made. We give an outline of this procedure here.

TABLE V
CONSTRAINED TEST PROBLEMS USED IN THIS STUDY

Problem	n	Variable bounds	Objective functions	Constraints
CONSTE	2	$x_1 \in [0.1, 1.0]$ $x_2 \in [0, 5]$	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = (1 + x_2)/x_1$	$g_1(\mathbf{x}) = x_2 + 9x_1 \geq 6$ $g_2(\mathbf{x}) = -x_2 + 9x_1 \geq 1$
SRN	2	$x_i \in [-20, 20]$ $i = 1, 2$	$f_1(\mathbf{x}) = (x_1 - 2)^2 + (x_2 - 1)^2 + 2$ $f_2(\mathbf{x}) = 9x_1 - (x_2 - 1)^2$	$g_1(\mathbf{x}) = x_1^2 + x_2^2 \leq 225$ $g_2(\mathbf{x}) = x_1 - 3x_2 \leq -10$
TNK	2	$x_i \in [0, \pi]$ $i = 1, 2$	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = x_2$	$g_1(\mathbf{x}) = -x_1^2 - x_2^2 + 1 + 0.1 \cos(16 \arctan(x_1/x_2)) \leq 0$ $g_2(\mathbf{x}) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5$
WATER	3	$0.01 \leq x_1 \leq 0.45$ $0.01 \leq x_2 \leq 0.10$ $0.01 \leq x_3 \leq 0.10$	$f_1(\mathbf{x}) = 106780.37(x_2 + x_3) + 61704.67$ $f_2(\mathbf{x}) = 3000x_1$ $f_3(\mathbf{x}) = (305700)2289x_2 / (0.06 \times 2289)^{0.65}$ $f_4(\mathbf{x}) = (250)2289 \exp(-39.75x_2 + 9.9x_3 + 2.74)$ $f_5(\mathbf{x}) = 25(1.39)/(x_1x_2) + 4940x_3 - 80$	$g_1(\mathbf{x}) = 0.00139/(x_1x_2) + 4.94x_3 - 0.08 \leq 1$ $g_2(\mathbf{x}) = 0.000306/(x_1x_2) + 1.082x_2 - 0.0986 < 1$ $g_3(\mathbf{x}) = 12.307/(x_1x_2) + 49408.24x_3 + 4051.02 \leq 50000$ $g_4(\mathbf{x}) = 2.098/(x_1x_2) + 8046.33x_3 - 606.71 \leq 16000$ $g_5(\mathbf{x}) = 2.138/(x_1x_2) - 7883.39x_3 - 705.04 \leq 10000$ $g_6(\mathbf{x}) = 0.417/(x_1x_2) + 1721.26x_3 - 136.54 \leq 2000$ $g_7(\mathbf{x}) = 0.164/(x_1x_2) + 631.13x_3 - 54.48 \leq 550$

All objective functions are to be minimized.

Three different nondominated rankings of the population are first performed. The first ranking is performed using M objective function values and the resulting ranking is stored in a N -dimensional vector R_{obj} . The second ranking R_{con} is performed using only the constraint violation values of all (J of them) constraints and no objective function information is used. Thus, constraint violation of each constraint is used as a criterion and a nondomination classification of the population is performed with the constraint violation values. Notice that for a feasible solution all constraint violations are zero. Thus, all feasible solutions have a rank 1 in R_{con} . The third ranking is performed on a combination of objective functions and constraint-violation values [a total of $(M + J)$ values]. This produces the ranking R_{com} . Although objective function values and constraint violations are used together, one nice aspect of this algorithm is that there is no need for any penalty parameter. In the domination check, criteria are compared individually, thereby eliminating the need of any penalty parameter. Once these rankings are over, all *feasible* solutions having the best rank in R_{com} are chosen for the new population. If more population slots are available, they are created from the remaining solutions systematically. By giving importance to the ranking in R_{obj} in the selection operator and by giving importance to the ranking in R_{con} in the crossover operator, the investigators laid out a systematic multi-objective GA, which also includes a niche-preserving operator. For details, readers may refer to [17]. Although the investigators did not compare their algorithm with any other method, they showed the working of this constraint-handling method on a number of engineering design problems. However, since nondominated sorting of three different sets of criteria are required and the algorithm introduces many different operators, it remains to be investigated how it performs on more complex

problems, particularly from the point of view of computational burden associated with the method.

In the following section, we choose a set of four problems and compare the simple constrained NSGA-II with the Ray-Tai-Seow's method.

C. Simulation Results

We choose four constrained test problems (see Table V) that have been used in earlier studies. In the first problem, a part of the unconstrained Pareto-optimal region is not feasible. Thus, the resulting constrained Pareto-optimal region is a concatenation of the first constraint boundary and some part of the unconstrained Pareto-optimal region. The second problem SRN was used in the original study of NSGA [20]. Here, the constrained Pareto-optimal set is a subset of the unconstrained Pareto-optimal set. The third problem TNK was suggested by Tanaka *et al.* [21] and has a discontinuous Pareto-optimal region, falling entirely on the first constraint boundary. In the next section, we show the constrained Pareto-optimal region for each of the above problems. The fourth problem WATER is a five-objective and seven-constraint problem, attempted to solve in [17]. With five objectives, it is difficult to discuss the effect of the constraints on the unconstrained Pareto-optimal region. In the next section, we show all $\binom{5}{2}$ or ten pairwise plots of obtained nondominated solutions. We apply real-coded NSGA-II here.

In all problems, we use a population size of 100, distribution indexes for real-coded crossover and mutation operators of 20 and 100, respectively, and run NSGA-II (real coded) with the proposed constraint-handling technique and with Ray-Tai-Seow's constraint-handling algorithm [17] for a maximum of 500 generations. We choose this rather large number of generations to investigate if the spread in solutions

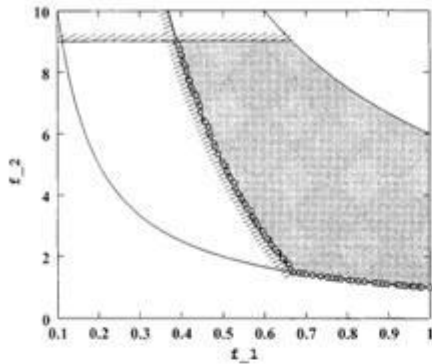


Fig. 14. Obtained nondominated solutions with NSGA-II on the constrained problem CONSTR.

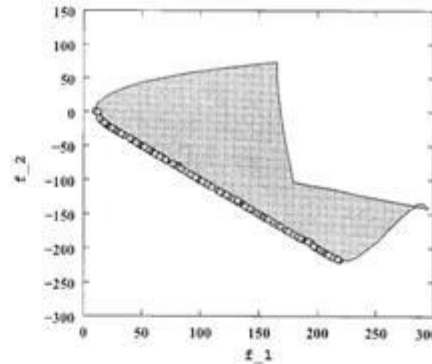


Fig. 16. Obtained nondominated solutions with NSGA-II on the constrained problem SRN.

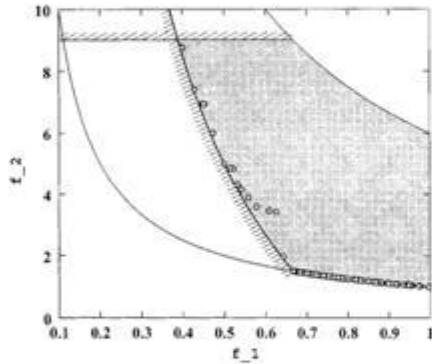


Fig. 15. Obtained nondominated solutions with Ray-Tai-Seow's algorithm on the constrained problem CONSTR.

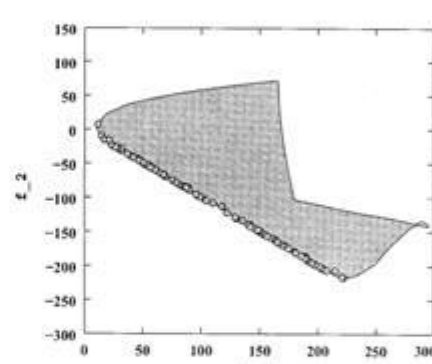


Fig. 17. Obtained nondominated solutions with Ray-Tai-Seow's algorithm on the constrained problem SRN.

can be maintained for a large number of generations. However, in each case, we obtain a reasonably good spread of solutions as early as 200 generations. Crossover and mutation probabilities are the same as before.

Fig. 14 shows the obtained set of 100 nondominated solutions after 500 generations using NSGA-II. The figure shows that NSGA-II is able to uniformly maintain solutions in both Pareto-optimal region. It is important to note that in order to maintain a spread of solutions on the constraint boundary, the solutions must have to be modified in a particular manner dictated by the constraint function. This becomes a difficult task of any search operator. Fig. 15 shows the obtained solutions using Ray-Tai-Seow's algorithm after 500 generations. It is clear that NSGA-II performs better than Ray-Tai-Seow's algorithm in terms of converging to the true Pareto-optimal front and also in terms of maintaining a diverse population of nondominated solutions.

Next, we consider the test problem SRN. Fig. 16 shows the nondominated solutions after 500 generations using NSGA-II.

The figure shows how NSGA-II can bring a random population on the Pareto-optimal front. Ray-Tai-Seow's algorithm is also able to come close to the front on this test problem (Fig. 17).

Figs. 18 and 19 show the feasible objective space and the obtained nondominated solutions with NSGA-II and Ray-Tai-Seow's algorithm. Here, the Pareto-optimal region is discontinuous and NSGA-II does not have any difficulty in finding a wide spread of solutions over the true Pareto-optimal region. Although Ray-Tai-Seow's algorithm found a number of solutions on the Pareto-optimal front, there exist many infeasible solutions even after 500 generations. In order to demonstrate the working of Fonseca-Fleming's constraint-handling strategy, we implement it with NSGA-II and apply on TNK. Fig. 20 shows 100 population members at the end of 500 generations and with identical parameter setting as used in Fig. 18. Both these figures demonstrate that the proposed and Fonseca-Fleming's constraint-handling strategies work well with NSGA-II.

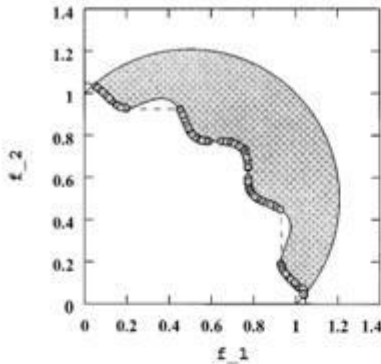


Fig. 18. Obtained nondominated solutions with NSGA-II on the constrained problem TNK.

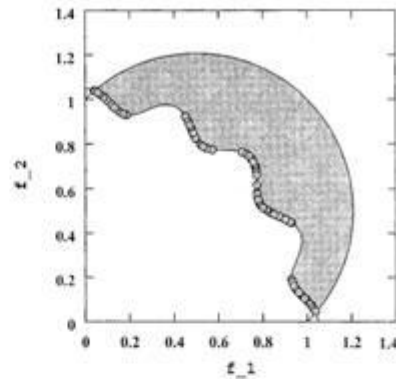


Fig. 20. Obtained nondominated solutions with Fonseca-Fleming's constraint-handling strategy with NSGA-II on the constrained problem TNK.

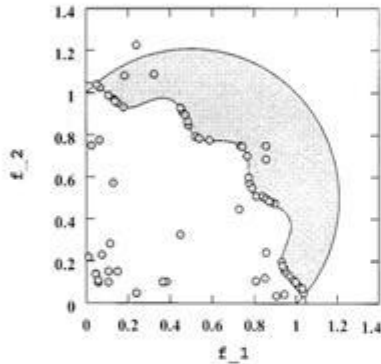


Fig. 19. Obtained nondominated solutions with Ray-Tai-Seow's algorithm on the constrained problem TNK.

Ray *et al.* [17] have used the problem WATER in their study. They normalized the objective functions in the following manner:

$$f_1/8(10^4), f_2/1500, f_3/3(10^6), f_4/6(10^6), f_5/8000.$$

Since there are five objective functions in the problem WATER, we observe the range of the normalized objective function values of the obtained nondominated solutions. Table VI shows the comparison with Ray-Tai-Seow's algorithm. In most objective functions, NSGA-II has found a better spread of solutions than Ray-Tai-Seow's approach. In order to show the pairwise interactions among these five normalized objective functions, we plot all $\binom{5}{2}$ or ten interactions in Fig. 21 for both algorithms. NSGA-II results are shown in the upper diagonal portion of the figure and the Ray-Tai-Seow's results are shown in the lower diagonal portion. The axes of any plot can be obtained by looking at the corresponding diagonal boxes and their ranges. For example, the plot at the first row and third column has its vertical axis as f_3 and horizontal axis as f_5 . Since this plot belongs in the upper side of the diagonal, this

plot is obtained using NSGA-II. In order to compare this plot with a similar plot using Ray-Tai-Seow's approach, we look for the plot in the third row and first column. For this figure, the vertical axis is plotted as f_3 and the horizontal axis is plotted as f_1 . To get a better comparison between these two plots, we observe Ray-Tai-Seow's plot as it is, but turn the page 90° in the clockwise direction for NSGA-II results. This would make the labeling and ranges of the axes same in both cases.

We observe that NSGA-II plots have better formed patterns than in Ray-Tai-Seow's plots. For example, figures f_1 - f_3 , f_1 - f_4 , and f_3 - f_4 interactions are very clear from NSGA-II results. Although similar patterns exist in the results obtained using Ray-Tai-Seow's algorithm, the convergence to the true fronts is not adequate.

VII. CONCLUSION

We have proposed a computationally fast and elitist MOEA based on a nondominated sorting approach. On nine different difficult test problems borrowed from the literature, the proposed NSGA-II was able to maintain a better spread of solutions and converge better in the obtained nondominated front compared to two other elitist MOEAs—PAES and SPEA. However, one problem, PAES, was able to converge closer to the true Pareto-optimal front. PAES maintains diversity among solutions by controlling crowding of solutions in a deterministic and pre-specified number of equal-sized cells in the search space. In that problem, it is suspected that such a deterministic crowding coupled with the effect of mutation-based approach has been beneficial in converging near the true front compared to the dynamic and parameterless crowding approach used in NSGA-II and SPEA. However, the diversity preserving mechanism used in NSGA-II is found to be the best among the three approaches studied here.

On a problem having strong parameter interactions, NSGA-II has been able to come closer to the true front than the other two approaches, but the important matter is that all three approaches faced difficulties in solving this so-called highly epistatic problem. Although this has been a matter of ongoing

TABLE VI
LOWER AND UPPER BOUNDS OF THE OBJECTIVE FUNCTION VALUES OBSERVED IN THE OBTAINED NONDOMINATED SOLUTIONS

Algorithm	f_1	f_2	f_3	f_4	f_5
NSGA-II	0.798 - 0.920	0.027 - 0.900	0.095 - 0.951	0.031 - 1.110	0.001 - 3.124
Ray-Tai-Seow	0.810 - 0.956	0.046 - 0.834	0.967 - 0.934	0.036 - 1.561	0.211 - 3.116

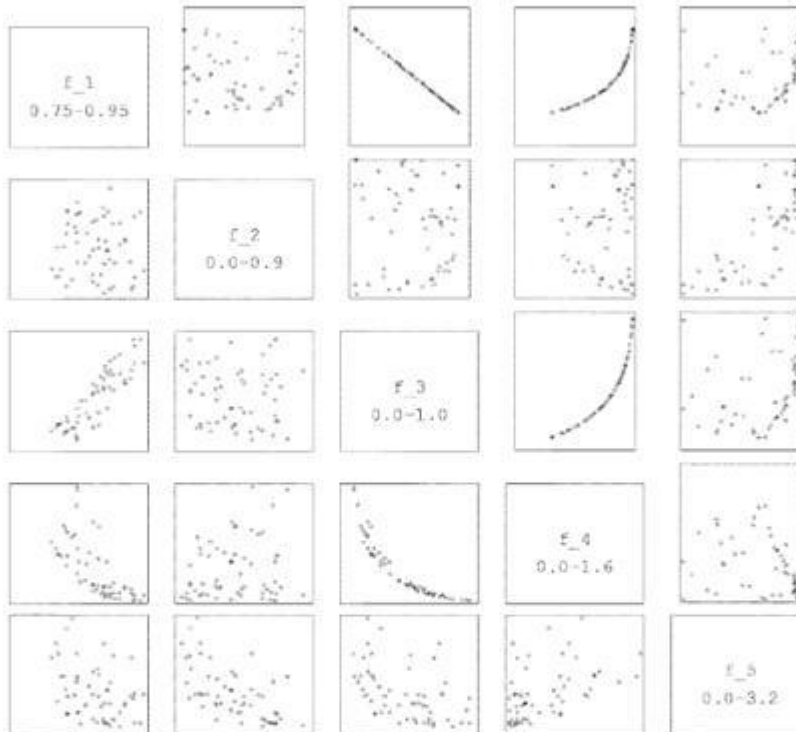


Fig. 21. Upper diagonal plots are for NSGA-II and lower diagonal plots are for Ray-Tai-Seow's algorithm. Compare (i, j) plot (Ray-Tai-Seow's algorithm with $i > j$) with (j, i) plot (NSGA-II). Label and ranges used for each axis are shown in the diagonal boxes.

research in single-objective EA studies, this paper shows that highly epistatic problems may also cause difficulties to MOEAs. More importantly, researchers in the field should consider such epistatic problems for testing a newly developed algorithm for multiobjective optimization.

We have also proposed a simple extension to the definition of dominance for constrained multiobjective optimization. Although this new definition can be used with any other MOEAs, the real-coded NSGA-II with this definition has been shown to solve four different problems much better than another recently-proposed constraint-handling approach.

With the properties of a fast nondominated sorting procedure, an elitist strategy, a parameterless approach and a simple yet efficient constraint-handling method, NSGA-II, should find increasing attention and applications in the near future.

REFERENCES

- [1] K. Deb, *Multiobjective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.
- [2] —, "An efficient constraint-handling method for genetic algorithms," *Comput. Methods Appl. Mech. Eng.*, vol. 186, no. 2-4, pp. 311-338, 2000.
- [3] —, "Multiobjective genetic algorithms: Problem difficulties and construction of test functions," in *Evol. Comput.*, 1999, vol. 7, pp. 205-230.
- [4] K. Deb and D. E. Goldberg, "An investigation of niche and species formation in genetic function optimization," in *Proceedings of the Third International Conference on Genetic Algorithms*, J. D. Schaffer, Ed. San Mateo, CA: Morgan Kaufman, 1989, pp. 42-50.
- [5] K. Deb and S. Agrawal, "Understanding interactions among genetic algorithm parameters," in *Foundations of Genetic Algorithms V*, W. Banzhaf and C. Reeves, Eds. San Mateo, CA: Morgan Kaufman, 1998, pp. 265-286.
- [6] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," in *Complex Syst.*, Apr. 1995, vol. 9, pp. 115-148.

- [7] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," in *Proceedings of the Fifth International Conference on Genetic Algorithms*, S. Forrest, Ed. San Mateo, CA: Morgan Kaufman, 1993, pp. 416-423.
- [8] —, "On the performance assessment and comparison of stochastic multiobjective optimizers," in *Parallel Problem Solving from Nature II*, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds. Berlin, Germany: Springer-Verlag, 1996, pp. 584-593.
- [9] —, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms—Part I: A unified formulation," *IEEE Trans. Syst., Man, Cybern. A*, vol. 28, pp. 26-37, Jan. 1998.
- [10] —, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms—Part II: Application example," *IEEE Trans. Syst., Man, Cybern. A*, vol. 28, pp. 38-47, Jan. 1998.
- [11] D. E. Goldberg, B. Korb, and K. Deb, "Messy genetic algorithms: Motivation, analysis, and first results," in *Complex Syst.*, Sept. 1989, vol. 3, pp. 93-530.
- [12] G. Harik, "Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms," Illinois Genetic Algorithms Lab., Univ. Illinois at Urbana-Champaign, Urbana, IL, IJGAL Rep. 97005, 1997.
- [13] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," in *Proceedings of the First IEEE Conference on Evolutionary Computation*, Z. Michalewicz, Ed. Piscataway, NJ: IEEE Press, 1994, pp. 82-87.
- [14] J. Knowles and D. Corne, "The Pareto archived evolution strategy: A new baseline algorithm for multiobjective optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation*. Piscataway, NJ: IEEE Press, 1999, pp. 98-105.
- [15] F. Kuesawe, "A variant of evolution strategies for vector optimization," in *Parallel Problem Solving from Nature*, H.-P. Schwefel and R. Männer, Eds. Berlin, Germany: Springer-Verlag, 1990, pp. 193-197.
- [16] C. Poloni, "Hybrid GA for multiobjective aerodynamic shape optimization," in *Genetic Algorithms in Engineering and Computer Science*, G. Winter, J. Periaux, M. Galan, and P. Cuesta, Eds. New York: Wiley, 1997, pp. 397-414.
- [17] T. Ray, K. Tai, and C. Seow, "An evolutionary algorithm for multiobjective optimization," *Eng. Optim.*, vol. 33, no. 3, pp. 399-424, 2001.
- [18] G. Rudolph, "Evolutionary search under partially ordered sets," Dept. Comput. Sci./LS11, Univ. Dortmund, Dortmund, Germany, Tech. Rep. CI-67-99, 1999.
- [19] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of the First International Conference on Genetic Algorithms*, J. J. Grefenstette, Ed. Hillsdale, NJ: Lawrence Erlbaum, 1987, pp. 93-100.
- [20] N. Srinivas and K. Deb, "Multiobjective function optimization using nondominated sorting genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221-248, Fall 1995.
- [21] M. Tanaka, "GA-based decision support system for multicriteria optimization," in *Proc. IEEE Int. Conf. Systems, Man and Cybernetics-2*, 1995, pp. 1556-1561.
- [22] D. Van Veldhuizen, "Multiobjective evolutionary algorithms: Classifications, analyzes, and new innovations," Air Force Inst. Technol., Dayton, OH, Tech. Rep. AFTI-DS-ENG-99-01, 1999.
- [23] D. Van Veldhuizen and G. Lamont, "Multiobjective evolutionary algorithm research: A history and analysis," Air Force Inst. Technol., Dayton, OH, Tech. Rep. TR-98-03, 1998.
- [24] E. Zitzler, "Evolutionary algorithms for multiobjective optimization: Methods and applications," Doctoral dissertation, ETH 1339K, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 1999.
- [25] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173-195, Summer 2000.
- [26] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—A comparative case study," in *Parallel Problem Solving From Nature, V*, A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds. Berlin, Germany: Springer-Verlag, 1998, pp. 292-301.



Kalyanmoy Deb (A'02) received the B.Tech degree in mechanical engineering from the Indian Institute of Technology, Kharagpur, India, 1985 and the M.S. and Ph.D. degrees in engineering mechanics from the University of Alabama, Tuscaloosa, in 1989 and 1991, respectively.

He is currently a Professor of Mechanical Engineering with the Indian Institute of Technology, Kanpur, India. He has authored or coauthored over 100 research papers in journals and conferences, a number of book chapters, and two books: *Multiobjective Optimization Using Evolutionary Algorithms* (Chichester, U.K.: Wiley, 2001) and *Optimization for Engineering Design* (New Delhi, India: Prentice-Hall, 1995). His current research interests are in the field of evolutionary computation, particularly in the areas of multicriterion and real-parameter evolutionary algorithms.

Dr. Deb is an Associate Editor of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and an Executive Council Member of the International Society on Genetic and Evolutionary Computation.



Amrit Pratap was born in Hyderabad, India, on August 27, 1979. He received the M.S. degree in mathematics and scientific computing from the Indian Institute of Technology, Kanpur, India, in 2001. He is working toward the Ph.D. degree in computer science at the California Institute of Technology, Pasadena, CA.

He was a member of the Kanpur Genetic Algorithms Laboratory. He is currently a Member of the Caltech Learning Systems Group. His current research interests include evolutionary computation, machine learning, and neural networks.



Sameer Agarwal was born in Bulandshahr, India, on February 19, 1977. He received the M.S. degree in mathematics and scientific computing from the Indian Institute of Technology, Kanpur, India, in 2000. He is working toward the Ph.D. degree in computer science at University of California, San Diego.

He was a Member of the Kanpur Genetic Algorithms Laboratory. His research interests include evolutionary computation and learning both in humans as well as machines. He is currently developing learning methods for learning by imitation.



T. Meyariwan was born in Haldia, India, on November 23, 1977. He is working toward the M.S. degree in chemistry from Indian Institute of Technology, Kanpur, India.

He is a Member of the Kanpur Genetic Algorithms Laboratory. His current research interests include evolutionary computation and its applications to biology and various fields in chemistry.

ANEXO 4 GRÁFICAS DE LAS FRONTERAS

