



**UNIVERSIDAD NACIONAL DEL SUR**

**TESIS DE DOCTOR EN INGENIERÍA QUÍMICA**

**Diseño de Sistemas de Instrumentación para Plantas Químicas**

**Mercedes del Carmen Carnero**

**BAHÍA BLANCA**

**ARGENTINA**

**2016**





**UNIVERSIDAD NACIONAL DEL SUR**

**TESIS DE DOCTOR EN INGENIERÍA QUÍMICA**

**Diseño de Sistemas de Instrumentación para Plantas Químicas**

**Mercedes del Carmen Carnero**

**BAHÍA BLANCA**

**ARGENTINA**

**2016**

## **PREFACIO**

Esta Tesis se presenta como parte de los requisitos para optar al grado Académico de Doctor en Ingeniería Química, de la Universidad Nacional del Sur y no ha sido presentada previamente para la obtención de otro título en esta Universidad u otra. La misma contiene los resultados obtenidos en investigaciones llevadas a cabo en el ámbito de la Planta Piloto de Ingeniería Química, dependiente del Departamento de Ingeniería Química de la Universidad Nacional del Sur y del CONICET y de la Facultad de Ingeniería de la Universidad Nacional de Río Cuarto, bajo la dirección de la Dra. Mabel Cristina Sánchez.

Mercedes del Carmen Carnero



UNIVERSIDAD NACIONAL DEL SUR  
Secretaría General de Posgrado y Educación  
Continua

La presente tesis ha sido aprobada el ..../..../..... , mereciendo la calificación de .....(.....)

## **Agradecimientos**

A mi Facultad de Ingeniería de la Universidad Nacional de Río Cuarto por su apoyo en la realización de mis estudios de postgrado y a las autoridades de la Universidad Nacional del Sur y de la Planta Piloto de Ingeniería Química (PLAPIQUI) por recibirme en sus instalaciones y permitirme desarrollar los cursos de postgrado y esta tesis doctoral.

Agradezco especialmente a mi directora de tesis, Dra. Mabel C. Sánchez, quien en todos estos años me ha acompañado, demostrando no sólo su valía en lo disciplinar sino sus valores humanos que hacen que me sienta orgullosa de haber sido dirigida por ella.

Manifiesto además mi enorme gratitud a mis compañeros de grupo por sus valiosos aportes en el desarrollo de este trabajo y porque han sabido, en ocasiones, alivianar mis tareas cotidianas en pos de la finalización del mismo.

A mi familia, mi agradecimiento absoluto, por su paciencia, su apoyo incondicional, su comprensión en todo momento y su aliento constante y cariñoso.

Un párrafo aparte, lo merece mi madre. A ella en especial está dedicada esta tesis. Ha sido, en más de un sentido, una madre inspiradora. Con su sabiduría innata, supo inculcarme el deseo de una formación académica y con su ejemplo de trabajadora incansable me enseñó que nada se logra sin esfuerzo.

## Resumen

En los últimos años se han producido numerosos avances en la industria química en áreas tales como: monitoreo de procesos, control basado en modelos, control estadístico multivariable, optimización en tiempo real, diagnóstico de fallas, etc. Sin embargo, existen aún varios tópicos claves que deben resolverse, entre ellos, el diseño de la red de instrumentación de una planta química. Ésta proporciona un conocimiento integral del estado actual de proceso que resulta indispensable para la toma de decisiones. En este contexto aparece la necesidad de desarrollar estrategias sistemáticas que permitan determinar para cada variable de proceso si será medida, y en tal caso, indicar cuántos sensores la medirán y cuáles serán sus características, a la vez que se optimiza un criterio de desempeño preestablecido. Este problema se conoce como Problema de Diseño de Redes de Sensores, PDRS.

Frecuentemente es necesario satisfacer restricciones en la calidad de las estimaciones sólo para un conjunto de variables claves. En este caso, se diseña una red de sensores llamada General, sin conocer a priori la cardinalidad del conjunto óptimo de instrumentos. Si bien se han utilizado diferentes modelos de selección óptima de sensores, siempre resulta un problema de optimización combinatorial sujeto a restricciones.

En esta tesis se han desarrollado e implementado metodologías de resolución del PDRS para plantas de proceso cuya operación se representa mediante sistemas linealizados de ecuaciones algebraicas. Las estrategias propuestas emplean motores de optimización híbridos con el fin de explotar las ventajas de los algoritmos estocásticos de búsqueda global y local. Dichas metodologías, si bien no garantizan optimalidad, ofrecen buenas soluciones en tiempos que no crecen exponencialmente con el tamaño de

la entrada. Además son lo suficientemente flexibles como para ser adaptadas a otras formulaciones del PDRS.

Inicialmente se investiga la resolución del diseño empleando nuevos algoritmos de búsqueda local inspirados en la heurística Búsqueda Tabú (BT). Los resultados alcanzados indican que éstos son menos robustos cuando la complejidad de las restricciones aumenta. También se estudia el comportamiento de un algoritmo ad-hoc híbrido basado en Búsqueda Dispersa, que combina una búsqueda global con técnicas de inspección por entornos diferentes a las empleadas en la BT, obteniéndose una mejor reproducibilidad de los resultados.

Además se analiza la aplicabilidad de los Algoritmos de Estimación de Distribuciones (AEDs), de orden creciente, para la resolución del PDRS. Si bien todos son capaces de hallar buenas soluciones, la complejidad adicional que implica considerar un modelo gráfico probabilístico de mayor orden no produce una mejora importante en la calidad y repetitividad de las mejores soluciones.

Por último se desarrolla una nueva metodología, denominada pPBIL-OS (*Parallel Population Based Incremental Learning – Strategic Oscillations*). Ésta consiste en un esquema de resolución híbrido que combina las ventajas del algoritmo poblacional PBIL, de orden uno, con la técnica de búsqueda local Oscilaciones Estratégicas, que trabaja en el marco de la BT. El procedimiento emplea un cierto número de subpoblaciones que evolucionan de manera independiente, pudiendo intercambiar información entre ellas con cierta probabilidad. Su desempeño, medido en función de: la calidad y reproducibilidad de las soluciones, la capacidad de resolver problemas de tamaño incremental y los requerimientos de cómputo, resulta muy satisfactorio.

## Abstract

In recent years many improvements have occurred in chemical industries in areas such as: process monitoring, model predictive control, statistical process control, real time optimization, etc. However there exist many key topics which should be solved, e.g., the design of the instrumentation network of a chemical plant. It provides a comprehensive knowledge about the current process state that is very important for the decision making process. In this context, the development of systematic strategies which determine if a process variable will be measured, and indicating this case the quantity of sensors and their features, and also satisfying a performance criterion, is essential. This problem is known as the Sensor Network Design Problem, PDRS.

Frequently it is necessary to fulfill some restrictions on the estimation quality of a set of key variables. In this case, a General Sensor Network is designed without knowing in advance the cardinality of the optimum set. Even though different models are used for the optimal selection of instruments, a combinatorial optimization problem subject to constraints always arises.

In this thesis, resolution methodologies of the PDRS are developed and implemented for process plants whose operation is represented by a linearized system of algebraic equations. The proposed strategies use hybrid optimization engines that take advantage of the combined use of global and local search stochastic algorithms. Even though these methodologies do not guarantee optimality, they provide good solutions using computation times that do not exponentially depend on the problem size. Furthermore they are flexible enough to be adopted for other formulations of the PDRS.

At first the research addresses the design problem using new local search algorithms which are inspired on the Tabu Search (BT). Results indicate that their robustness



decreases when the restrictions complexity increases. Furthermore the behaviour of an ad-hoc hybrid algorithm based on Scatter Search is studied. It combines a global search with local search techniques which are different from the ones used by BT. A better solution reproducibility is achieved.

Also the applicability of Estimation of Distribution Algorithms (AEDs) of incremental order for the resolution of the PDRS is analyzed. Although they are capable to obtain good solutions, the additional complexity involved in considering a graphical probabilistic model of high order does not improve the solutions quality and reproducibility.

At last, a new methodology, called pPBIL-OS (Parallel Population Based Incremental Learning – Strategic Oscillations), is developed. It is a hybrid solving scheme than combines the advantages of the global search algorithm PBIL, of order one, with the local search technique Strategic Oscillations. It works in the framework of the BT. The procedure employs a certain number of subpopulations, which evolve independently and exchange information among them with a given probability. The strategy performance, which is evaluated in terms of the solutions quality and reproducibility, the capability to solve problems of incremental size and the computation time requirements, is very satisfactory.

## Índice General

<b>CAPÍTULO 1</b> .....	<b>1</b>
<b>1.1 Introducción</b> .....	<b>2</b>
<b>1.2 Diseño Óptimo de Redes Generales de Sensores para Monitoreo</b> .....	<b>6</b>
1.2.1 Formulación del Problema .....	6
1.2.2 Métodos de Solución Existentes.....	8
1.2.2.1. Algoritmos Exactos.....	8
1.2.2.2. Heurísticas.....	10
<b>1.3 Objetivos de la Tesis</b> .....	<b>12</b>
<b>1.4 Organización de la Tesis</b> .....	<b>14</b>
<b>1.5 Nomenclatura</b> .....	<b>15</b>
<b>1.6 Acrónimos</b> .....	<b>16</b>
<b>1.7 Referencias</b> .....	<b>16</b>
<b>CAPÍTULO 2</b> .....	<b>20</b>
<b>2.1 Introducción</b> .....	<b>21</b>
<b>2.2 Revisión de las metodologías existentes</b> .....	<b>21</b>
2.2.1 Algoritmos Exactos.....	26
2.2.1.1. Técnicas de Búsqueda en Árbol .....	26
2.2.1.2. Técnicas de Programación Matemática. ....	34
2.2.2 Heurísticas.....	39
<b>2.3 Conclusiones</b> .....	<b>43</b>
<b>2.4 Nomenclatura</b> .....	<b>45</b>
<b>2.5 Acrónimos</b> .....	<b>46</b>
<b>2.6 Referencias</b> .....	<b>47</b>

<b>CAPÍTULO 3.....</b>	<b>52</b>
<b>3.1 Introducción.....</b>	<b>53</b>
<b>3.2 Formulación del problema.....</b>	<b>54</b>
3.2.1 Restricciones de Estimabilidad.....	55
3.2.2 Restricciones de Precisión.....	58
<b>3.3 Nuevas Estrategias de Diseño de Redes de Sensores basadas en</b>	
<b>Búsqueda Tabú .....</b>	<b>59</b>
3.3.1 Algoritmo basado en Búsqueda Tabú Clásica.....	60
3.3.1.1. Generación de la Solución Inicial.....	60
3.3.1.2. Definición de la Vecindad.....	65
3.3.1.3. Función de Evaluación.....	65
3.3.1.4. Memorias de corto y largo plazo .....	66
3.3.1.5. Criterios de Aspiración y Detención.....	67
3.3.2 Algoritmo basado en la Búsqueda Tabú Clásica y la Técnica de	
Reencadenamiento de Caminos.....	68
3.3.2.1. Movimiento M1 .....	70
3.3.2.2. Movimiento M2.....	71
<b>3.4 Nuevas Estrategias de Diseño de Redes de Sensores basadas en Búsqueda</b>	
<b>Dispersa .....</b>	<b>72</b>
3.4.1 Algoritmo basado en Búsqueda Dispersa .....	73
3.4.1.1. Generación-Diversificación .....	73
3.4.1.2. Mejora de Soluciones.....	73
3.1.1.1 Generación y Actualización del Conjunto de Referencia .....	75
3.1.1.2 Generación de Subconjuntos y Combinación de Soluciones .....	75
<b>3.5 Resultados de Aplicación .....</b>	<b>77</b>

3.5.1 Descripción del Proceso.....	78
3.5.2 Casos de Estudio.....	80
3.5.3 Resultados .....	81
3.1.1.3 Algoritmo basado en BTC.....	81
3.1.1.4 Algoritmo basado en BTC-RC .....	83
3.1.1.5 Algoritmo basado en BD.....	84
<b>3.2 Conclusiones.....</b>	<b>85</b>
<b>3.3 Nomenclatura.....</b>	<b>86</b>
<b>3.4 Acrónimos .....</b>	<b>89</b>
<b>3.5 Referencias .....</b>	<b>90</b>
<b>CAPÍTULO 4.....</b>	<b>92</b>
<b>4.1 Introducción.....</b>	<b>93</b>
<b>4.2 Algoritmos de Estimación de Distribuciones .....</b>	<b>94</b>
4.2.1 Modelos Univariados .....	97
4.2.2 Modelos Bivariados .....	98
4.2.3 Modelos Multivariados .....	100
4.2.3.1. AEDs que utilizan Modelos de Factorizaciones de Productos Marginales.....	101
4.2.3.2. AEDs que utilizan Modelos de Redes Bayesianas .....	107
<b>4.3 Resultados .....</b>	<b>110</b>
4.3.1 Resultados de la aplicación del Algoritmo UMDA.....	111
4.3.2 Resultados de la aplicación de Algoritmos con aprendizaje estructural	112
<b>4.4 Conclusiones.....</b>	<b>113</b>
<b>4.5 Nomenclatura.....</b>	<b>118</b>
<b>4.6 Acrónimos .....</b>	<b>119</b>

<b>4.7 Referencias .....</b>	<b>121</b>
<b>CAPÍTULO 5.....</b>	<b>124</b>
<b>5.1 Introducción.....</b>	<b>125</b>
<b>5.2 Algoritmo de Estimación de Distribuciones con Aprendizaje</b>	
<b>Incremental .....</b>	<b>126</b>
<b>5.3 Oscilación Estratégica.....</b>	<b>129</b>
<b>5.4 Nuevo Algoritmo Híbrido de Diseño de Redes de Sensores .....</b>	<b>131</b>
5.4.1 Función de Evaluación.....	131
5.4.2 Inicialización del Vector de Probabilidades.....	132
5.4.3 Muestreo.....	135
5.4.4 Búsqueda Local .....	136
5.4.5 Pseudocódigo.....	143
<b>5.5 Ejemplos y Resultados .....</b>	<b>144</b>
<b>5.6 Conclusiones.....</b>	<b>147</b>
<b>5.7 Nomenclatura.....</b>	<b>147</b>
<b>5.8 Acrónimos .....</b>	<b>150</b>
<b>5.9 Referencias .....</b>	<b>151</b>
<b>CAPÍTULO 6.....</b>	<b>152</b>
<b>6.1 Introducción.....</b>	<b>153</b>
<b>6.2 Caso 1 .....</b>	<b>155</b>
<b>6.3 Caso 2 .....</b>	<b>161</b>
<b>6.4 Caso 3 .....</b>	<b>166</b>
<b>6.5 Conclusiones.....</b>	<b>175</b>
<b>6.6 Nomenclatura.....</b>	<b>177</b>

<b>6.7 Acrónimos .....</b>	<b>178</b>
<b>6.8 Referencias.....</b>	<b>178</b>
<b>CAPÍTULO 7.....</b>	<b>180</b>
<b>7.1 Conclusiones.....</b>	<b>181</b>
<b>7.2 Trabajos futuros .....</b>	<b>184</b>
<b>7.3 Acrónimos .....</b>	<b>188</b>

## Índice de Tablas

Tabla 3.1. Datos del SMN.....	79
Tabla 3.2. Parámetros del algoritmo basado en BTC .....	81
Tabla 3.3. Datos y Resultados para el Diseño 1 - BTC .....	82
Tabla 3.4. Estadísticos de la solución para el Diseño 1 - BTC .....	82
Tabla 3.5. Datos y Resultados para el Diseño 2 – BTC.....	83
Tabla 3.6. Estadísticos de la solución para el Diseño 2 - BTC .....	83
Tabla 3.7. Parámetros del algoritmo basado en BTC-RC.....	83
Tabla 3.8. Estadísticos de la solución para el Diseño 1 – BTC-RC .....	84
Tabla 3.9. Estadísticos de la solución para el Diseño 2 – BTC-RC .....	84
Tabla 3.10. Parámetros del algoritmo basado en BD .....	85
Tabla 3.11. Estadísticos de la solución para el Diseño 1 – BD.....	85
Tabla 3.12. Estadísticos de la solución para el Diseño 2 – BD.....	85
Tabla 4.1. Estadísticos de la solución para el Diseño 1 – UMDA .....	112
Tabla 4.2. Estadísticos de la solución para el Diseño 2 – UMDA .....	112
Tabla 4.3. Estadísticos de la solución para el Diseño 1 – AffEDA .....	112
Tabla 4.4. Estadísticos de la solución para el Diseño 2 – AffEDA .....	112
Tabla 4.5. Estadísticos de la solución para el Diseño 1 – EBNA.....	113
Tabla 4.6. Estadísticos de la solución para el Diseño 2 – EBNA.....	113
Tabla 5.1. Parámetros del algoritmo pPBIL-OS.....	145
Tabla 5.2. Datos y Resultados para el Diseño 1 - pPBIL-OS .....	145

Tabla 5.3. Estadísticos de la solución para el Diseño 1 - pPBIL-OS .....	145
Tabla 5.4. Datos y Resultados para el Diseño 2 - pPBIL-OS .....	146
Tabla 5.5. Estadísticos de la solución para el Diseño 2 - pPBIL-OS .....	146
Tabla 6.1. Especificaciones de los diseños – Caso 1 .....	155
Tabla 6.2. Resultados obtenidos para D3 - pPBIL-OS, AffEDA y EBNA.....	156
Tabla 6.3. Resultados obtenidos para D4 - pPBIL-OS, AffEDA y EBNA.....	157
Tabla 6.4. Estadísticos de los tiempos de ejecución – D3 .....	159
Tabla 6.5. Estadísticos de los tiempos de ejecución – D4 .....	160
Tabla 6.6. Datos - Caso 2.....	161
Tabla 6.7. Especificaciones del diseño – Caso 2.....	163
Tabla 6.8. Resultados obtenidos para el Caso 2 - pPBIL-OS, AffEDA y EBNA.....	164
Tabla 6.9. Estadísticos de los tiempos de ejecución - Caso 2 .....	166
Tabla 6.10. Datos – Caso 3 .....	167
Tabla 6.11. Especificaciones del diseño - Caso 3.....	170
Tabla 6.12. Resultados obtenidos para el Caso 3 - pPBIL-OS, AffEDA y EBNA.....	170
Tabla 6.13. Estadísticos de los tiempos de ejecución - Caso 3 .....	173



## Índice de Figuras

Figura 1.1: Paradigma de la Planta Inteligente.....	2
Figura 3.1. Pseudocódigo del algoritmo basado en BTC.....	68
Figura 3.2. Pseudocódigo para el algoritmo de RC.....	72
Figura 3.3. Pseudocódigo para el algoritmo de Búsqueda Dispersa.....	77
Figura 3.4. Diagrama de Flujo del SMN.....	78
Figura 4.1: Esquema general de funcionamiento de un AED.....	96
Figura 4.2. Pseudocódigo para un AED.....	96
Figura 4.3. Pseudocódigo para ECGA.....	102
Figura 4.4. Pseudocódigo para AffEDA.....	105
Figura 4.5. Ejemplo de grupos obtenidos empleando la técnica AP.....	105
Figura 4.6. Ejemplo de estimación de parámetros para AffEDA.....	106
Figura 4.7.: Esquema general para el algoritmo BOA.....	108
Figura 4.8: Red Bayesiana a partir de la población seleccionada.....	109
Figura 4.9: Estimación de parámetros: Evaluación de las probabilidades condicionales .....	110
Figura A4.1: Esquema básico de un modelo gráfico probabilístico.....	116
Figura 5.1: Pseudocódigo de PBIL.....	127
Figura 5.2. Pseudocódigo de pPBIL usando un procedimiento de mejora.....	129
Figura 5.3. Pseudocódigo para el algoritmo OS.....	131
Figura 5.4: Construcción de la población inicial $D_0$ a partir de un conjunto previo PP .....	135

Figura 5.5. Estimación del vector de probabilidades inicial $p^0$ .....	135
Figura 5.6. Generación de $\mathcal{S}^1$ a partir de $p^0$ .....	136
Figura 5.7. Pseudocódigo para la generación de la vecindad $\mathcal{N}_i(\mathbf{q})$ .....	138
Figura 5.8. Pseudocódigo del algoritmo OS propuesto para resolver el PDRS .....	139
Figura 5.9: Agrupamiento y mejora utilizando el algoritmo OS .....	141
Figura 5.10. Recorrido de las soluciones en las fases constructivas y destructivas del algoritmo OS .....	142
Figura 5.11. Recorrido de las soluciones factibles y no factibles en el algoritmo OS	142
Figura 5.12. Pseudocódigo para el algoritmo pPBIL-OS .....	144
Figura 6.1. Mejores valores encontrados para D3 - pPBIL-OS, AffEDA y EBNA .....	156
Figura 6.2. Mejores valores encontrados para D4 - pPBIL-OS, AffEDA y EBNA .....	158
Figura 6.3. Tiempos de Ejecución para D3 - pPBIL-OS, AffEDA y EBNA.....	159
Figura 6.4. Tiempos de Ejecución para D4 - pPBIL-OS, AffEDA y EBNA.....	160
Figura 6.5. Diagrama de flujo – Caso 2 .....	163
Figura 6.6. Mejores valores encontrados para el Caso 2 - pPBIL-OS, AffEDA y EBNA .....	165
Figura 6.7. Tiempos de Ejecución para el Caso 2 - pPBIL-OS, AffEDA y EBNA .....	166
Figura 6.8. Mejores valores encontrados para el Caso 3 - pPBIL-OS, AffEDA y EBNA .....	172
Figura 6.9. Tiempos de Ejecución para el Caso 3 - pPBIL-OS, AffEDA y EBNA .....	173
Figura 6.10. Mejores valores encontrados para el Caso 3 - pPBIL-OS, AffEDA.....	174
Figura 6.11. Estadísticos de la función de evaluación – Caso 3 - pPBIL-OS y AffEDA .....	175



# ***Capítulo 1***

**Marco General**

### 1.1 Introducción

En la última década se evidencia la tendencia a integrar el control de los procesos, la operación de toda la planta y el sistema de negocios de la corporación en tiempo real, lo cual es posible gracias a los rápidos avances en la infraestructura cibernética y la tecnología de las comunicaciones. La denominada Planta Inteligente, esquematizada en la Figura 1.1 (Christofides y col., 2007), tiene por objetivo maximizar el rédito económico de la actividad con responsabilidad social, cumpliendo normas estrictas sobre cuidado del medio ambiente, salud ocupacional y seguridad.

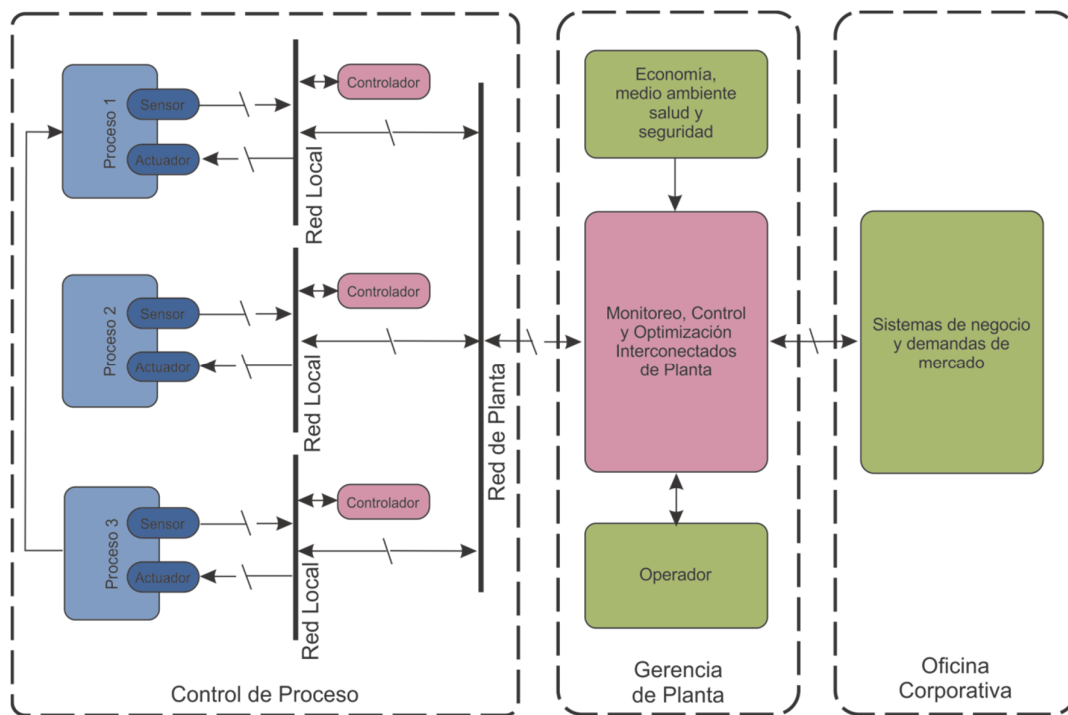


Figura 1.1. Paradigma de la Planta Inteligente

Siguiendo el paradigma de la Planta Inteligente, cada activo ejecuta su función de proceso básica a la vez que proporciona información actual y predictiva sobre su desempeño, mediante redes de comunicación en tiempo real, a la gerencia de la planta. Por su parte, ésta se encarga de maximizar el uso (y en consecuencia el valor) de cada

activo en base al desempeño conjunto, actual y futuro, de todos ellos, ante el cambio permanente de las condiciones de los negocios. A fin de concretar esta visión, la Planta Inteligente consta de tres niveles diferentes a los que nos referiremos como: Procesos, Gerencia de Planta y Oficina Corporativa.

En el nivel Procesos, las redes de áreas locales manejan el control retroalimentado de las unidades de proceso en tiempo real. Dichas redes se unen a la Red de Planta para intercambiar información, por ejemplo, valores de las mediciones y *set-points* de los controladores. En el nivel Gerencia de Planta se fijan las condiciones operativas en base a consideraciones económicas, restricciones relacionadas con el medio ambiente, salud ocupacional y seguridad, y los requerimientos de producción, determinados por la Oficina Corporativa teniendo en cuenta las demandas del mercado y las estrategias de los negocios. Entre estos niveles el flujo de información no es jerárquico, sino que la mayoría de la información es compartida entre ellos.

Si bien se han producido numerosos avances en áreas tales como diseño de controladores basados en modelos, optimización en tiempo real, control supervisor, monitoreo de procesos, control estadístico multivariable, etc., se reconoce que para convertir en realidad el paradigma de la Planta Inteligente existen aún varios tópicos claves que deben resolverse. Entre ellos, Christofides y col. (2007) mencionan el diseño de la red de sensores, ya que ésta proporciona un conocimiento integral del estado actual de proceso, el cual resulta indispensable para la toma de decisiones. Dichos autores resaltan la necesidad de desarrollar estrategias sistemáticas que aborden la determinación de las variables medidas, su localización óptima, la frecuencia y el tiempo de la medición, como así también, la especificación de la precisión, tiempo medio hasta la falla y condiciones de mantenimiento de los instrumentos utilizados. Además prevén que los

futuros desarrollos se realizarán tomando como base las estrategias existentes basadas en consideraciones y métricas de lazo abierto.

El planeamiento de un sistema de instrumentación es una tarea compleja de múltiples niveles, que comprende la definición de los objetivos globales, la selección de las variables medidas y la especificación de los detalles de implementación. La calidad de la información disponible del proceso depende esencialmente de la selección realizada en el segundo nivel. Por un lado, el grado de estimabilidad de las variables es función exclusivamente de la topología de la planta y de las observaciones seleccionadas (caudal, presión, temperatura, composición, etc.). En tanto que, la precisión, exactitud y disponibilidad de las estimaciones dependen de los mismos factores, así como también de las características particulares de los instrumentos instalados (precisión y confiabilidad) y de las tareas de mantenimiento previstas durante la vida útil del sistema de instrumentación.

Aunque se pueden identificar varios miles de variables de estado en un proceso químico, sólo un subconjunto de ellas se mide debido a causas económicas y de factibilidad técnica. La selección del subconjunto de variables medidas, se realiza durante la preparación del Diagrama de Proceso e Instrumentación. Es práctica común basar la decisión en la experiencia previa con plantas similares y en reglas empíricas, ya que no existen paquetes de cómputo que ayuden al diseñador en esta tarea. En consecuencia el desarrollo de estrategias para efectuar la selección óptima de sensores resulta de sumo interés, ya que permitirá realizar una asignación óptima de los recursos económicos, que asegure la disponibilidad de la información requerida del proceso y niveles de seguridad aceptables.

La localización de sensores comprende determinar para cada variable de proceso si será medida, y en tal caso, indicar cuántos sensores la medirán y cuáles son sus características (por ejemplo: costo, precisión, probabilidad de falla, etc.). Frecuentemente es necesario satisfacer restricciones de calidad sólo para un conjunto de variables denominadas claves. En este caso, se diseña una red de sensores llamada General, sin conocer a priori la cardinalidad del conjunto óptimo de instrumentos. Si bien se han utilizado diferentes modelos de selección óptima de sensores, siempre resulta un problema de optimización combinatorial sujeto a restricciones.

El diseño óptimo de redes de sensores (RS) con fines de Control Regulatorio de Procesos (Seborg y col., 1989; Ogunnaike y Ray, 1994; Luyben y Floudas, 1994 a,b) y Análisis de Falla (Bhushan y Rengaswamy, 2002; Bhushan y col., 2008; Rodríguez y col., 2013 y 2016) se caracteriza por emplear modelos dinámicos, cuantitativos, cualitativos y basados en datos históricos, correspondientes a la operación de un equipo o sectores de planta. Si el interés es lograr información más integral del proceso, a fin de monitorear variables relacionadas, por ejemplo, con la cuantificación de la producción, la calidad del producto, el desempeño operativo de equipos, la optimización en línea del proceso, el impacto sobre el medio ambiente, etc., la representación de su operación en estado estacionario es la alternativa más adecuada, a fin de evitar la complejidad inherente al empleo de modelos dinámicos en plantas de gran escala. Cabe aclarar que la presente tesis se focaliza en la localización óptima de sensores con el fin específico de monitorear un conjunto de variables claves asociadas con la operación de un proceso en estado estacionario.



## 1.2 Diseño Óptimo de Redes Generales de Sensores para Monitoreo

### 1.2.1 Formulación del Problema

Se supone que la operación de una planta en estado estacionario se representa mediante el vector de ecuaciones algebraicas no lineales

$$\mathbf{R}(\mathbf{z}) = \mathbf{R}(\mathbf{x}, \mathbf{u}) = \mathbf{0}, \quad (1.1)$$

siendo  $\mathbf{z}$  el vector  $I$  dimensional de variables de proceso, y  $\mathbf{x}$  y  $\mathbf{u}$  los vectores de variables medidas y no medidas respectivamente. La selección óptima de instrumentos durante la etapa de diseño de la planta consiste en determinar la partición óptima del vector  $\mathbf{z}$  en los vectores  $\mathbf{x}$  y  $\mathbf{u}$ , y de manera general se representa mediante la siguiente ecuación

$$\begin{aligned} & \text{Min / Max} && f(\mathbf{q}) \\ & \text{s.t.} && \\ & && \mathbf{g}(\mathbf{q}) \leq \mathbf{g}^*(\mathbf{q}) \\ & && \mathbf{q} \in \{0,1\}^I \end{aligned} \quad (1.2)$$

donde  $\mathbf{q}$  es el vector  $I$  dimensional de variables binarias, tal que:  $q_i = 1$  si la  $i$ -ésima variable se mide, y  $q_i = 0$  en caso contrario (Bagajewicz, 1997). Además  $f(\mathbf{q})$  y  $\mathbf{g}(\mathbf{q})$  indican el criterio de optimización y el vector de restricciones impuestas sobre la calidad de las estimaciones de las variables claves, respectivamente.

En general se obtiene un problema de optimización discreto no lineal. La dimensión del espacio de búsqueda aumenta significativamente para procesos de gran escala, por lo que el diseño involucra la resolución de un problema de optimización combinatorial grande, que en general, presenta varios óptimos locales. En tales casos, resulta valioso que la estrategia de solución proporcione al menos una buena solución, aunque no sea la óptima, y pueda ejecutarse empleando computación en paralelo a fin de reducir el tiempo de cómputo.

En los últimos años se ha extendido el estudio y aplicación de una formulación particular del Problema de Diseño de Redes de Sensores (PDRS) representado por la ecuación (1.2). Ésta corresponde a la minimización del costo de adquisición del sistema de instrumentación sujeta a restricciones de: estimabilidad y precisión de las estimaciones, obtenidas utilizando estrategias de Reconciliación de Datos, de las variables claves. Este PDRS se simboliza mediante la siguiente ecuación

$$\begin{aligned}
 & \text{Min} && \mathbf{c}^T \mathbf{q} \\
 & \text{s.t.} && \\
 & && E_{le}(\mathbf{q}) \geq 1 && \forall le \in \mathcal{S}_e && (1.3) \\
 & && \hat{\sigma}_{lp}(\mathbf{q}) \leq \sigma_{lp}^* && \forall lp \in \mathcal{S}_\sigma \\
 & && \mathbf{q} \in \{0,1\}^I
 \end{aligned}$$

en la cual  $\mathbf{c}^T$  es el vector de costos de adquisición de los instrumentos,  $E_{le}$  representa el Grado de Estimabilidad de la variable  $le$  (Bagajewicz y Sánchez, 1999),  $\hat{\sigma}_{lp}$  indica el Desvío Estándar de la Estimación de la variable  $lp$  obtenido luego de aplicar estrategias de Reconciliación de Datos, y  $\sigma_{lp}^*$  es su correspondiente límite superior. Las restricciones de estimabilidad y precisión se imponen sobre las variables pertenecientes a los subconjuntos  $\mathcal{S}_e$  y  $\mathcal{S}_\sigma$ , siendo  $\mathcal{S}_\sigma$  un subconjunto de  $\mathcal{S}_e$ .

En esta formulación el límite inferior de  $E_{le}$  se fija en uno, por lo cual sólo se requiere la ejecución de un algoritmo de clasificación de variables (Sánchez y Romagnoli, 1996) para verificar la factibilidad de una solución. Se asume que el conjunto de ecuaciones algebraicas lineales, obtenido por linealización del sistema  $\mathbf{R}(\mathbf{x}, \mathbf{u}) = \mathbf{0}$  alrededor del punto operativo de estado estacionario, representa la operación de la planta. Además se considera que las mediciones presentan errores aleatorios no correlacionados.

## 1.2.2 Métodos de Solución Existentes

El PDRS abarca una amplia gama de formulaciones con diferentes funciones objetivo: algunas de tipo económico y otras donde se incluyen medidas de desempeño tales como tolerancia a fallas, confiabilidad asociada a la red, etc. Respecto del conjunto de restricciones, la situación es similar. Esto da como resultado diferentes tipos de problemas a resolver, con características particulares cada uno de ellos y que a priori sólo comparten el hecho de poder ser clasificados dentro del conjunto de problemas de optimización combinatoria. Puede decirse también que, salvo algunas formulaciones particulares, la mayoría presentan una complejidad de resolución alta.

Las técnicas de solución utilizadas para este tipo de problemas pueden ser clasificadas, de manera general, en dos grandes grupos, los cuales se mencionan a continuación.

### 1.2.2.1 Algoritmos Exactos

Los Algoritmos Exactos son aquellos que garantizan la obtención de óptimos globales. Hasta la actualidad se han presentado diversos algoritmos comprendidos en esta categoría con el fin de resolver el PDRS. Sin embargo no se han reportado métodos de resolución que garanticen la obtención del óptimo global para cualquier tamaño de una instancia en tiempo polinomial. Dentro de este enfoque se distinguen:

- a) Técnicas de búsqueda en árbol: Son estrategias de enumeración inteligente del conjunto de soluciones, aplicadas a problemas de optimización combinatoria, que intentan evitar la inspección de todo el conjunto de soluciones posibles. Una implementación de esta técnica es la forma general del método Ramificación y Acotación (*Branch and Bound*, B&B). En principio sólo se requiere que un conjunto de soluciones, representadas

por un nodo, pueda partitionarse en subconjuntos mutuamente excluyentes. Cada uno de estos representa un hijo del nodo original. El término ramificación se refiere a estas particiones sucesivas del espacio de soluciones. Además se debe contar con un algoritmo para el cálculo de un límite inferior sobre el valor de la función objetivo de cualquier solución en un subconjunto dado. Estos límites inferiores (acotación) se utilizan para construir una prueba de optimalidad sin inspección exhaustiva de todo el espacio de búsqueda (Papadimitriou y Steiglitz, 1982). El alcance de esta metodología está dado por el tamaño de la instancia a resolver. Los problemas de gran escala consumirían tiempos de cómputo prohibitivos.

- b) Técnicas de Programación Matemática: si el diseño puede formularse como un problema de optimización planteado en términos de una función objetivo y un conjunto de restricciones definidas de manera explícita, y asimilarse a una formulación predeterminada, por ejemplo, a un problema de tipo mezcla entera no lineal (*Mixed Integer Nonlinear Programming, MINLP*), entonces es posible utilizar paquetes de optimización dedicados y resolver el problema mediante técnicas, tales como, B&B, Rama y Corte (*Branch and Cut*) (Nemhauser y Wolsey, 1999), Método de Descomposición de Benders Generalizada (Geoffrion, 1972) o el Método de las Aproximaciones Exteriores (Duran y Grossmann, 1986; Fletcher y Leyfer, 1994), entre otros. En cuanto a su aplicación, adolecen de las mismas limitaciones que las técnicas previamente citadas.

En la literatura publicada en los últimos años sobre el PDRS, el mayor tamaño de instancia resuelta empleando Algoritmos Exactos corresponde a problemas con aproximadamente 50 variables.

### 1.2.2.2. Heurísticas

Son estrategias que, si bien no garantizan optimalidad, se utilizan ampliamente para resolver problemas complejos y de gran escala en diferentes áreas. Existe en la literatura un gran número de algoritmos asociados a este tipo de enfoque de solución. Algunos son de tipo poblacional, otros se basan exclusivamente en búsquedas por entorno, y además se han presentados métodos híbridos.

En relación con los métodos poblacionales, los Algoritmos Evolutivos, AEs, en general, (Michalewicz, 1996) y los Algoritmos Genéticos, AGs, (Holland, 1975) en particular se han desarrollado, probado y analizado con el fin de resolver diferentes clases de problemas conocidos como NP-completos durante las últimas décadas. Desde el punto de vista de la teoría de optimización, son métodos probabilísticos de orden cero, ya que sólo requieren los valores de la función a ser optimizada, y de búsqueda global, dado que manejan un conjunto de potenciales soluciones, llamado población, que evolucionan iterativamente a través del mecanismo de selección y operadores de cruzamiento y mutación. Son capaces de salir de un óptimo local dado que mantienen selectivamente un conjunto de soluciones de menor desempeño que la actual, las cuales pueden evolucionar hacia el óptimo global. Su desventaja es que, dada su naturaleza estocástica, no ofrecen ninguna garantía de convergencia para una corrida dada. La bondad de los mismos necesita ser validada a través de un cuidadoso análisis estadístico basado en un diseño de experiencias adecuado.

Además, su desempeño puede verse afectado por los valores asignados a la probabilidad de mutación, probabilidad de cruzamiento, tamaño de la población, tamaño de la población seleccionada y número de individuos reemplazados en cada iteración. Por otra parte, los AGs simples no tienen una memoria adicional, ni mecanismos previstos para preservar las buenas estructuras genotípicas en el proceso de búsqueda. Más aún,

uno de los principales operadores evolutivos, el cruzamiento, puede ejercer una acción disruptiva durante la evolución ocasionando la pérdida de soluciones promisorias. Asimismo, el empleo de la selección como único nexo entre el problema a resolver (resumido en la función de evaluación) y el algoritmo en sí, puede no ser suficiente para explorar el entorno de buenas soluciones.

La necesidad de resolver los inconvenientes presentados por los AGs clásicos ha originado el desarrollo de otras técnicas evolutivas, tales como los Algoritmos de Estimación de Distribuciones, AEDs, (Larrañaga y Lozano, 2002; Hauschild y Pelikan 2011). Su idea central es que el proceso evolutivo lo sufre un modelo probabilístico expresado en términos de la función de distribución de probabilidad conjunta de las variables aleatorias que participan del modelo. En esta propuesta, el conjunto de soluciones del problema a resolver es el resultado del muestreo aleatorio del modelo probabilístico.

Las ventajas que ofrece este enfoque en relación con los AGs, es que el operador de cruzamiento deja de existir, y el proceso evolutivo es guiado por sucesivas estimaciones del modelo que, al ser muestreado, proporciona un conjunto de “buenas” soluciones. La construcción de este modelo debería, en principio, tener en cuenta la interrelación entre las variables del problema, de manera tal de favorecer la permanencia de las “buenas estructuras”, es decir, la combinación de caracteres que son comunes a la mayoría de las mejores soluciones.

Por otra parte, los algoritmos de búsqueda local se destinan a la inspección intensiva de buenas regiones del espacio de búsqueda. Dentro de esta categoría se encuentra la heurística Búsqueda Tabú (BT), desarrollada inicialmente por Glover y Laguna (1997). Ésta es básicamente una técnica de inspección de mejores soluciones por

entorno, que registra o guarda en una memoria temporal las regiones ya visitadas. Se ha reportado su aplicación para la resolución de problemas de optimización relativos a la Ingeniería Química (Cavin y col. 2004; Lin y Miller, 2004 a-b).

También existe un amplio espectro de estrategias híbridas, entre las cuales puede mencionarse la Búsqueda Dispersa (BD). Ésta consiste en la evolución de una población mediante la combinación de dos o más soluciones y la mejora de las mismas utilizando diferentes procedimientos de búsqueda local. La BD ha sido aplicada a diferentes problemas de optimización discretos de reconocida complejidad.

De manera general las metaheurísticas pueden definirse como “procesos maestros iterativos que guían y modifican las operaciones de heurísticas subordinadas para producir de manera eficiente soluciones de alta calidad. Pueden manipular una única solución completa (o incompleta) o una colección de soluciones en cada iteración. Las heurísticas subordinadas pueden ser procedimientos de alto o bajo nivel, una simple búsqueda local, o un método de construcción" (Voss y col., 1999). Las metaheurísticas son algoritmos aproximados, por lo general no determinísticos, y en principio independientes del problema, aunque pueden hacer uso del conocimiento específico del mismo en forma de heurísticas que son controladas por la estrategia del nivel superior. Su empleo ha permitido obtener soluciones de calidad para numerosos problemas muy complejos (Blum y col., 2011; Urli, 2014).

### **1.3 Objetivos de la Tesis**

La motivación de esta tesis está basada en las limitaciones encontradas para resolver los PDRS Generales correspondientes a plantas de tamaño considerable. Se ha demostrado que la eficiencia de los métodos exactos, que aseguran encontrar una solución

global al problema de diseño óptimo, se reduce notablemente con el aumento de escala. Esta limitación proviene fundamentalmente de la naturaleza combinatoria del problema, que resulta en la imposibilidad de contar con algoritmos de tiempo polinomial para los casos de interés práctico. Los métodos de optimización heurísticos surgen entonces como la alternativa de solución más factible para abordar diseños de mayor dimensión.

En consecuencia, los objetivos de esta tesis son desarrollar e implementar metaheurísticas para abordar el diseño óptimo de RS Generales dedicadas al monitoreo de plantas, cuya operación se representa mediante sistemas lineales o linealizados de ecuaciones algebraicas. Se propone utilizar esquemas híbridos con el fin de explotar las ventajas de los algoritmos estocásticos de búsqueda global y local. Con tal fin, la estrategia maestra propuesta para la resolución del PDRS es un AED, capaz de manejar simultáneamente un conjunto de soluciones, mientras que el procedimiento subordinado es una heurística particular de búsqueda local. Entre ellas, se ha seleccionado la BT por su buen desempeño para la resolución de problemas que involucran decisiones sobre la estructura y la secuencia de procesos en Ingeniería Química. Se pondrá especial énfasis en desarrollar técnicas que incorporen el conocimiento provisto por las ecuaciones de balance para acelerar la búsqueda.

Cabe aclarar que todos los ejemplos de aplicación presentados en esta tesis corresponden a procesos cuya operación se ha descrito, por simplicidad, mediante conjuntos de ecuaciones de balance de masa total. Sin embargo, la aplicación de las metodologías desarrolladas a plantas químicas representadas mediante modelos no lineales, luego de linealizarlos alrededor del punto de operación de estado estacionario, es directa.



## 1.4 Organización de la Tesis

A continuación se presenta una breve descripción del contenido de cada uno de los capítulos de la tesis.

El presente capítulo contiene las motivaciones y objetivos de las tareas de investigación desarrolladas y proporciona una visión general sobre los enfoques usados para la resolución de los PDRS Generales.

En el Capítulo 2 se presenta una revisión crítica de las metodologías existentes para resolver, en forma automática, el problema de diseño óptimo de instrumentación.

El PDRS que se aborda en esta tesis se formula en el Capítulo 3, y en él se detallan los procedimientos utilizados para el cálculo de las restricciones. Luego se presentan estrategias de resolución del PDRS de diferente tipo. Las primeras son metodologías de optimización de búsqueda local basadas en BT y la restante es una técnica híbrida que emplea BD y BT.

En el Capítulo 4 se describen los aspectos fundamentales de los AED. Se los categoriza en términos de su capacidad para describir modelos que puedan captar la interrelación entre las variables y se analiza la complejidad que supone su implementación.

La resolución del PDRS planteado, empleando un AED dotado de búsqueda local, se aborda en el Capítulo 5. La metaheurística desarrollada involucra un AED de orden uno como estrategia maestra, y un mecanismo de Oscilación Estratégica en el marco de la BT como heurística subordinada.

El Capítulo 6 presenta ejemplos de PDRS para plantas de diferentes tamaños y distintos grados de especificación en las restricciones, cuyos resultados permiten validar la robustez y escalabilidad del método propuesto en el Capítulo 5.

Las conclusiones y futuras líneas de investigación se analizan en el Capítulo 7.

## 1.5 Nomenclatura

<b>c</b>	Vector de costos de instrumentación
$E_{le}$	Grado de estimabilidad de la variable $le$
$f$	Función objetivo
<b>g</b>	Vector de restricciones impuestas
$I$	Número de variables de proceso
<b>q</b>	Vector de variables binarias
<b>R</b>	Vector de ecuaciones algebraicas no lineales
$\mathcal{S}_\sigma$	Conjunto de variables de proceso sujetas a restricciones de precisión
$\mathcal{S}_\varepsilon$	Conjunto de variables de proceso sujetas a restricciones de estimabilidad
<b>u</b>	Vector de variables no medidas
<b>x</b>	Vector de variables medidas
<b>z</b>	Vector de variables

## Letras Griegas

$\hat{\sigma}_{lp}$	Desvío estándar de la estimación de la $lp$ -ésima variable
$\sigma_{lp}^*$	Desvío estándar superior para la estimación de la $lp$ -ésima variable

**1.6 Acrónimos**

AE	Algoritmo Evolutivo
AED	Algoritmo de Estimación de Distribuciones
AG	Algoritmo Genético
B&B	Algoritmo de <i>Branch and Bound</i>
BT	Búsqueda Tabú
BD	Búsqueda Dispersa
MINLP	<i>Mixed Integer non Linear Programming</i>
PDRS	Problema de Diseño de Redes de Sensores
RS	Redes de Sensores

**1.7 Referencias**

- Bagajewicz, M. Design and Retrofit of Sensor Networks in Process Plants. *AIChE J.*, **43**, 2300-2306 (1997).
- Bagajewicz, M., Sánchez, M. Cost Optimal Design and Upgrade of Non-Redundant and Redundant Linear Sensor Networks. *AIChE J.*, **45**, 1927-1938 (1999).

- Bhushan, M., Rengaswamy, R. Comprehensive Design of a Sensor Network for Chemical Plants based on Various Diagnosability and Reliability Criteria 1 Framework. *Industrial and Engineering Chemistry Research*, **41**, 1826-1839 (2002).
- Bhushan, M., Narasimhan, S., Rengaswamy, R. Robust sensor network design for fault diagnosis. *Computers and Chemical Engineering*, **32** (4-5), 1067-1084 (2008).
- Blum, C., Puchinger, J., Raidl, G., Roli. A. Hybrid metaheuristics in combinatorial optimization: A survey. *Applied Soft Computing*, **11**, 4135–4151(2011).
- Cavin, L., Fischer, U., Glover, F.; Hungerbühler, K. Multiobjective Process Design in Multi-purpose Batch Plants using a Tabu Search Optimization Algorithm. *Comp. Chem. Eng.*, **28**, 459-478 (2004).
- Christofides, P. D., Davis, J. F., El-Farra, N. H., Clark, D., Harris, K. R. D. , Gipson, J. N. Smart plant operations: Vision, progress and challenges. *AIChE J.*, **53**,2 734–2741 (2007).
- Duran, M.A. , Grossmann, I.E. An Outer Approximation Algorithm for a Class of Mixed Integer Non Linear Programs. *Mathematical Programming*, **36**, 307. (1986).
- Fletcher, R., Leyffer, S. Solving Mixed Integer Non Linear Programs by Outer Approximation. *Mathematical Programming*, **66**,327. (1994)
- Geoffrion, A.M. Generalized Benders Decomposition. *Journal of Optimization Theory and Applications*, **10**(4), 237-260 (1972).
- Glover, F.; Laguna, M. *Tabu Search*. Kluwer Academic Publishers: Norwell, MA, (1997).
- Hauschild, M., Pelikan, M. An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation*, **1**, 111–128 (2011).
- Holland, J., *Adaptation in Natural and Artificial Systems*, University of Michigan, USA (1975).
- Larrañaga, P., Lozano, J. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Boston, MA: Kluwer (2002).

- Lin, B.; Miller, D.C. Solving Heat Exchanger Network Synthesis Problems with Tabu Search. *Comp. Chem. Eng.*, **28**, 1451-1464 (2004).
- Lin, B.; Miller, D.C. Tabu Search Algorithm for Chemical Process Optimization. *Comp. Chem. Eng.*, **28**, 2287-2306 (2004).
- Luyben, M., Floudas, C. Analyzing the interaction of design and control-1. A multiobjective framework and application to binary distillation synthesis. *Computers and Chemical Engineering*, **18**, 933-969 (1994a).
- Luyben, M.L., Floudas, C. Analyzing the interaction of design and control-2. Reactor-separator-recycle system. *Computers and Chemical Engineering*, **18**, 971-994 (1994b).
- Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, Berlin Heidelberg, New York (1996).
- Nemhauser, G. and L. Wolsey, *Integer and Combinatorial Optimization*, Wiley Interscience, New York (1999).
- Ogunaike, B., Ray, W. *Process Dynamic, Modeling and Control*, Oxford University Press, New York (1994).
- Papadimitriou, C., and K. Steiglitz, *Combinatorial Optimization – Algorithms and Complexity*, Prentice Hall, New Jersey (1982).
- Rodríguez, L., Cedeño, M., Sánchez, M. A Structural Approach to Design Sensor Networks for Fault Diagnosis”, *Industrial and Engineering Chemistry Research*, **52**, 17941-17952 (2013).
- Rodríguez, L., Cedeño, M., Sánchez, M. Optimal Sensor Network Upgrade for Fault Detection using Principal Component Analysis. Aceptado para su publicación en *Industrial and Engineering Chemistry Research*, 11/02/2016.
- Sánchez, M., Romagnoli, J. Use of Orthogonal Transformations in Classification/Data Reconciliation. *Computers and Chemical Engineering*, **20**, 483-493 (1996).
- Seborg, D.E., Edgar, T., Mellichamp, D. *Process Dynamics and Control*, John Wiley, New York (1989).

Urli, T. *Hybrid Meta-Heuristics for Combinatorial Optimization*. Ph.D. Thesis. Università degli Studi di Udine Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica. Dottorato di Ricerca in Ingegneria Industriale e dell'Informazione, (2014).

Voss, S., Osman, I., Roucairol, C. *Meta-heuristics: Advances and trends in local search paradigms for optimization*. Kluwer Academic Publishers Norwell, MA, USA (1999).

# ***Capítulo 2***

---

## **Revisión Bibliográfica**

## 2.1 Introducción

En las plantas químicas modernas resulta imprescindible conocer los valores de diferentes variables y parámetros del proceso con determinada precisión y confiabilidad, a fin de realizar tareas de diagnóstico de la operación, control, optimización en línea, mantenimiento basado en la confiabilidad, etc.

Afortunadamente no es necesario efectuar la medición directa de cada una de las incógnitas (variables requeridas o claves), lo cual sería una práctica irrealizable para algunas de ellas. Si se miden un subconjunto de variables seleccionadas estratégicamente, las restantes pueden calcularse utilizando los balances de masa y energía.

En ciertas aplicaciones existen diferentes configuraciones de sensores cuya localización permite satisfacer el grado de conocimiento requerido del proceso. En tal caso, se pretende encontrar la configuración óptima de sensores que satisface uno o varios criterios, tales como: el costo total de instrumentación, el error global de estimación, la confiabilidad del sistema, etc. La determinación del conjunto óptimo de instrumentos se conoce como Problema de Diseño de Redes de Sensores (PDRS).

En este capítulo se incluye una revisión del estado actual del conocimiento en este tema. Posteriormente se presentan las conclusiones de la revisión bibliográfica que determinaron los objetivos que condujeron la investigación.

## 2.2 Revisión de las metodologías existentes

El problema de ubicación de sensores en plantas químicas está estrechamente asociado al de la clasificación de las variables de un proceso. Dada una planta que opera en estado estacionario, sus variables están relacionadas fundamentalmente por las leyes



de conservación de masa y energía. Por razones de costo, conveniencia o factibilidad técnica no se miden todas las variables del proceso. No obstante, algunas variables no medidas pueden ser estimadas utilizando los valores de las mediciones mediante la resolución de ecuaciones de balance de masa total, por componentes, balances de energía, etc. La posibilidad de estimar las variables no medidas depende fundamentalmente de la estructura del diagrama de flujo del proceso y de la ubicación de los instrumentos. Normalmente se dispone de un conjunto incompleto de mediciones, de modo que las variables no medidas se clasifican en observables y no observables. Las primeras se pueden estimar por cálculos a partir de los valores de las mediciones. En cuanto a las mediciones, éstas se dividen en redundantes y no redundantes. Una medición es redundante, si en caso de no efectuarse dicha medición la variable resulta determinable. La Figura 2.1 esquematiza la categorización de variables.

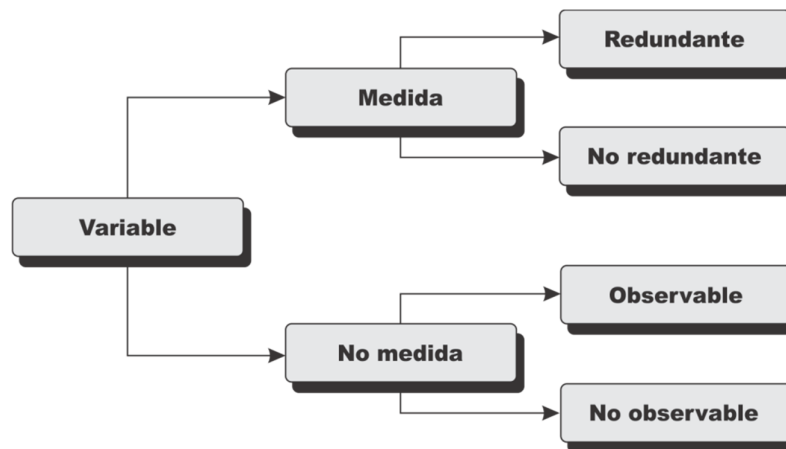


Figura 2.1 Clasificación de variables de proceso

En la literatura se presentan diferentes algoritmos de clasificación de variables, algunos se basan en la teoría de grafos y otros están orientados al uso de ecuaciones (Mah, 1990; Madron y Veverka, 1992; Romagnoli y Sánchez, 2000; Narasimhan y Jordache, 2000).

El diseño de instrumentación asistido con algoritmos de clasificación de variables para procesos cuya operación se describe mediante un sistema de ecuaciones algebraicas es un proceso iterativo. Dado un conjunto inicial de sensores, el diseñador aplica la estrategia de clasificación de variables a fin de determinar si las variables requeridas son estimables. Si esta condición se satisface, se procede luego a verificar el cumplimiento de otras restricciones, por ejemplo de precisión o confiabilidad de las estimaciones. Si resulta necesario modificar el conjunto de instrumentos, el diseñador decide la incorporación y/o eliminación de sensores en base a su conocimiento del proceso y juicio ingenieril. Las etapas de clasificación de variables, análisis de los resultados de la clasificación y modificación del conjunto de sensores se repiten secuencialmente hasta completar la selección de los instrumentos que permiten conocer toda la información deseada del proceso. Cabe destacar que este procedimiento no involucra la optimización de criterios de tipo económico en forma explícita, aunque se tiende a reducir la cantidad de instrumentos instalados.

Hacia fines de la década del 80 aparecen los primeros trabajos de DRS con fines diferentes a los de control y análisis de fallas. El desarrollo de estrategias de Reconciliación de Datos generó una revisión de los criterios utilizados para el diseño de estructuras de sensores, ya que se dispuso de una herramienta que permitió mejorar la precisión de las estimaciones de las variables de proceso utilizadas con fines de cuantificación del inventario, análisis de desempeño de equipos, modelado y optimización en línea, etc.

A fin de sintetizar los aportes de diferentes investigadores en la resolución del PDRS, se analizarán los estudios específicos sobre diseño de Redes Generales. Una red de sensores (RS) se define como General si está formada por un número de instrumentos desconocido de antemano, que surge de optimizar la condición de desempeño de la red,

sujeta a restricciones asociadas con un conjunto de variables claves. A diferencia de las RS Mínima y Redundante, la observabilidad de todas las variables no medidas no es una condición necesaria.

El diseño de redes de sensores con fines de monitoreo de procesos fue abordado inicialmente por Vaclavek y Loucka (1976), quienes propusieron el uso de la clasificación de las variables contenidas en los balances de masa y multicomponentes como herramienta para diseñar una RS que satisficiera la observabilidad de un conjunto de variables requeridas. Utilizaron un programa de clasificación de variables en sistemas bilineales, basado en la teoría de grafos.

Posteriormente Madron y Veverka (1992) presentaron una estrategia que consiste, inicialmente, en dividir la matriz de ocurrencia del proceso considerando las variables no medidas y medidas, separadas en requeridas y no requeridas, y ordenar las variables no medidas en orden creciente de factibilidad técnica y económica de medición. Luego se aplica un procedimiento de eliminación de Gauss-Jordan sobre la matriz ordenada. Si el análisis de la matriz resultante indica que no todas las variables requeridas son medidas u observables, se seleccionan nuevos sensores que corresponden a las variables más fáciles de medir. Los autores demostraron que la solución es óptima, considerando al costo como función objetivo, si no existen variables no observables no requeridas. En caso contrario no puede asegurarse la optimalidad de la solución, aunque ésta resulta una buena solución práctica. Madron y Veverka propusieron dos funciones objetivos: el costo de instrumentación y el desvío estándar promedio de las estimaciones de las variables requeridas, calculadas aplicando procedimientos de Reconciliación de Datos.

A continuación Luong y col. (1994) desarrollaron una estrategia de localización de sensores que permite la estimación única de un conjunto de variables requeridas, y a la

vez disponer de la estimación para un subconjunto de ellas aún en el caso de falla de  $i$  sensores, con el mínimo costo de instrumentación. El método aplica los conceptos enunciados por Mah y col. (1976) para la categorización de las variables intervinientes en sistemas lineales mediante el análisis de los ciclos del grafo del proceso. Este concepto también fue empleado por Meyer y col. (1994) para analizar la estimabilidad de un conjunto de variables requeridas, como parte de un procedimiento de diseño de sistemas de monitoreo de mínimo costo, para plantas representadas por balances lineales. Se consigue la solución óptima mediante un método tipo Ramificación y Acotación (*Branch and Bound*, B&B). El límite inferior de la rama, luego de la inclusión de  $i$  sensores en la red, es la suma del costo de instrumentación para el nivel  $i$  del árbol de búsqueda más el costo del instrumento más barato que podría incorporarse a cada ciclo fundamental del grafo, para conseguir la observabilidad de las variables requeridas. El límite superior es el costo de la primera solución encontrada y su valor se actualiza durante el procedimiento.

Bagajewicz (1997) fue el primero en formular el PDRS en términos de un vector de variables binarias  $\mathbf{q}$  de dimensión  $I$ , tal que:  $q_i = 1$  si la  $i$ -ésima variable se mide y  $q_i = 0$  en caso contrario. A partir de este trabajo, el empleo de variables binarias para indicar la presencia o ausencia de una medición se generalizó. Se han presentado numerosas formulaciones del PDRS Generales en la literatura durante los últimos quince años, que difieren en la función objetivo y las restricciones de los problemas de optimización propuestos. La característica común de todos ellos es su naturaleza combinatorial, por lo que su solución resulta una tarea compleja a medida que aumenta el tamaño del proceso considerado.

Desde 1997 hasta la actualidad se han desarrollado dos tipos de enfoques a fin de resolver los PDRS Generales planteados, los cuales se analizan a continuación.

### 2.2.1 Algoritmos Exactos

Las metodologías exactas garantizan la obtención del óptimo global. Dentro de esta categoría se encuentran dos tipos de estrategias:

#### 2.2.1.1. Técnicas de Búsqueda en Árbol

Son estrategias del tipo B&B, sin relajación en los nodos, que emplean formas inteligentes de exploración de árboles mediante la definición de cotas y criterios de parada adecuados.

Bagajewicz (1997) analizó el PDRS General en el cual se minimiza el costo de adquisición de la red sujeto a restricciones sobre la precisión de las estimaciones de las variables, obtenidas al aplicar la estrategia de Reconciliación de Datos, para un proceso cuya operación se representa mediante el sistema de ecuaciones algebraicas lineales asociado al balance de masa total. Considerando que hay un solo tipo de instrumento disponible para medir cada caudal másico, el PDRS se formuló de la siguiente manera:

$$\begin{aligned}
 & \text{Min} && \mathbf{c}^T \mathbf{q} \\
 & \text{s.t.} && \\
 & && \hat{\sigma}_{l_p}(\mathbf{q}) \leq \sigma_{l_p}^* \quad \forall l_p \in \mathcal{S}_\sigma \\
 & && \mathbf{q} \in \{0,1\}^I
 \end{aligned} \tag{2.1}$$

siendo  $\mathbf{c}^T$  el vector de costos de adquisición de instrumentos,  $\hat{\sigma}_{l_p}$  el Desvío Estándar de la Estimación de la variable  $l_p$ , perteneciente al conjunto de variables claves con restricciones de precisión  $\mathcal{S}_\sigma$ , y  $\sigma_{l_p}^*$  su correspondiente límite superior.

Además se consideraron como restricciones otras medidas de desempeño de la red tales como: la detectabilidad de errores gruesos, la resiliencia a los errores gruesos no detectados, y la precisión de las estimaciones ante fallas de sensores.

Los problemas de optimización resultantes son del tipo Mezcla Entera No Lineal (*Mixed Integer Non-Linear Programing*, MINLP), dado que las restricciones son funciones no lineales del vector de variables binarias  $\mathbf{q}$ . Por otra parte, las medidas de desempeño propuestas no se expresaron de manera explícita en términos de  $\mathbf{q}$ , lo cual impidió el uso de técnicas de solución tradicionales del tipo B&B con relajación en los nodos.

Para darle una solución al problema manteniendo la formulación previa, Bagajewicz (1997) desarrolló una estrategia de búsqueda en árbol con criterio de parada. El árbol de soluciones de la búsqueda en anchura propuesta se muestra en la Figura 2.2 y presenta las siguientes características:

- a) el nodo  $\mathbf{q}=\mathbf{0}$  es no factible;
- b) inicialmente los nodos alcanzados mediante el agregado de mediciones corresponden a soluciones no factibles dado que no se satisface la observabilidad de las variables claves o su precisión; a medida que se agregan más mediciones se alcanzan nodos factibles; el nivel del nodo para el cual se consigue la factibilidad de la solución varía de una rama a otra;
- c) suponiendo que todos los caudales másicos se miden con el mismo tipo de sensor, cuyo costo y precisión son independientes de su medida, el costo total de instrumentación incrementa de un nivel al próximo, y el/los primer/os nodo/s factible/s de una rama es/son el/los de menor costo del árbol que ésta genera;
- d) suponiendo que hay más de un tipo de sensor posible para efectuar todas las mediciones y que cada tipo de medidor de caudal másico tiene un costo y precisión que no dependen de su tamaño, el costo de cada nodo tiene un límite inferior que corresponde a la suma de los costos de los instrumentos más baratos que podrían emplearse para medir los flujos másicos asociados al nodo.

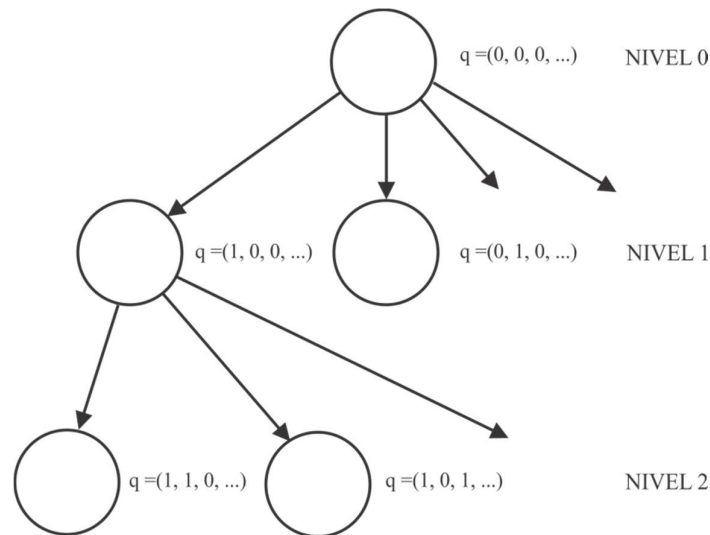


Figura 2.2 Árbol de soluciones de la búsqueda en anchura

Sobre la base de estas características, Bagajewicz propuso el siguiente esquema de enumeración en árbol para resolver el PDRS:

- 1) Partir del nodo raíz ( $\mathbf{q=0}$ )
- 2) Construir el árbol mediante la técnica de búsqueda en anchura. Una rama se desarrolla hasta que se cumple el criterio de parada. Luego volver un nivel hacia arriba y desarrollar la próxima rama. En cada nodo resolver el problema de optimización. El criterio de parada indica que si existe un solo tipo de instrumento para medir todas las corrientes, la exploración de una rama debe detenerse en el nivel correspondiente al de los primeros nodos factibles. Si existe más de un tipo de instrumento para medir todas las corrientes y el costo del nodo factible es superior a su límite inferior, entonces es necesario continuar la búsqueda unos pocos niveles más abajo. Dicho número de niveles se calcula en función de la mejor solución encontrada, el menor costo de esta solución y el costo del instrumento más barato. El criterio de parada evita que la búsqueda sea exhaustiva y en consecuencia poco práctica.

El procedimiento presentado por Bagajewicz (1997) es eficiente sólo para sistemas pequeños. Por tal motivo Gala y Bagajewicz (2006a) reemplazaron los nodos del árbol, que originalmente estaban formados por la unión de mediciones, por la unión de conjuntos de corte del grafo del proceso. Un conjunto de corte se asocia a los caudales máxicos involucrados en un balance de masa formulado alrededor de determinadas unidades de proceso. El esquema de enumeración en árbol con búsqueda en profundidad propuesto por estos autores es el siguiente:

- 1) Encontrar todos los conjuntos de corte del grafo del proceso.
- 2) Eliminar las variables claves presentes en los conjuntos de corte. Los conjuntos resultantes se denominan bloques constructivos.
- 3) Formar además un bloque constructivo por cada variable clave.
- 4) Clasificar todos los bloques constructivos en orden ascendente según sus costos. El costo de un bloque constructivo es la suma de los costos de los instrumentos que intervienen en el mismo.
- 5) Partir del nodo raíz, caracterizado por su infactibilidad trivial, y construir el árbol mediante la técnica de búsqueda en profundidad. Desarrollar cada rama incorporando un bloque constructivo por vez, seleccionado del conjunto de bloques constructivos remanentes mediante el Criterio de Formación de Ramas. Éste indica que se incorpora el bloque constructivo cuya unión con los activos produce el menor costo de instrumentación. Este criterio se aplica indistintamente en la generación de nodos descendientes o hermanos.
- 6) El desarrollo de una rama se continúa hasta alcanzar el Criterio de Parada. Este indica que la exploración de esta parte del árbol se detiene, puesto que la incorporación de nuevos bloques constructivos siempre aumenta el costo de la RS.



- 7) Si durante la aplicación del Criterio de Formación de Ramas, se obtienen nodos que corresponden a conjuntos de mediciones previamente analizados, no se continúa con el análisis del nodo.

El procedimiento desarrollado por Gala y Bagajewicz (2006a) fue aplicado a procesos que comprenden hasta 11 unidades y 24 corrientes. Se reportaron los mismos óptimos globales que para el procedimiento anterior y el tiempo de cómputo empleado fue significativamente menor. Posteriormente los mismos autores (Gala y Bagajewicz, 2006b) presentaron técnicas para reducir el gasto computacional del algoritmo con el fin de abordar problemas de mayor tamaño. Propusieron un método de enumeración en árbol similar al anterior, pero redujeron el número de conjuntos de corte mediante una descomposición del grafo del proceso. Además modificaron las operaciones que se aplican en un nodo, incorporando en algunos casos la suma anillo entre conjuntos de corte.

Para la resolución de problemas de diseño no lineales, Nguyen y Bagajewicz (2008) propusieron tres métodos de búsqueda en árbol y compararon su desempeño en problemas de diferente tamaño y con distintos grados de especificaciones. Los métodos mencionados son los siguientes:

- a) Búsqueda en árbol hacia adelante basado en combinaciones de ecuaciones.
- b) Búsqueda en árbol hacia adelante basado en combinaciones de ecuaciones pero con descomposición del grafo bipartito original en subgrafos.
- c) Búsqueda hacia atrás basado en una lista de mediciones, considerando en el nodo inicial todas las variables medidas.

Las distintas estrategias se aplicaron a tres ejemplos de diferente tamaño. El primero, conocido como *Continuous Stirred Tank Reactor* (CSRT), comprende 13

variables (Bhusan y Rengaswamy, 2000). El segundo, citado como *Mineral Flotation Process*, MFP, consta de 24 variables (Smith y Ichiyen, 1973). El último, denominado *Tennessee Eastman Process*, TEP, involucra 47 variables (Ricker y Lee, 1995). En todos los casos se consideraron grados de especificaciones bajo, moderado y alto. Las soluciones obtenidas se validaron empleando un método de búsqueda hacia adelante, desarrollado anteriormente por los autores, en los ejemplos en que el tamaño del problema lo permitió.

Del análisis de los resultados alcanzados surge que la búsqueda hacia atrás basada en una lista de mediciones funciona aceptablemente cuando el diseño involucra altos grados de requerimientos, esto es: un gran número de variables claves, y por lo tanto la solución óptima exhibe también un gran número de variables medidas. En estos casos ninguna de las dos primeras técnicas es aplicable desde el punto de vista de los tiempos de cómputo. El TEP con alto grado de especificaciones se resuelve en un tiempo aceptable utilizando dicha estrategia.

Para el TEP con bajo grado de especificación, la búsqueda con descomposición del grafo bipartito original (segunda técnica) proporciona soluciones óptimas, verificadas por otros métodos, en un tiempo del orden de 2 minutos.

En relación con los resultados obtenidos para el TEP con mediano grado de especificación, se observa que: el segundo método proporciona soluciones subóptimas luego de 74 hs de cómputo, es decir, no se pudo garantizar que la solución encontrada sea un óptimo global y la tercera estrategia presenta un desempeño aún más pobre. Por su parte, la primera metodología resulta inaplicable para todos los casos del TEP.

Con posterioridad Nguyen y Bagajewicz (2011a) resolvieron el caso de estudio MFP de 24 variables para diferentes diseños con grados de especificación bajo, moderado

y alto. Los autores consideraron que los PDRS para el MFP con un grado de especificación alto eran los más complejos de resolver. Se propuso un método exacto basado en una técnica híbrida que combina búsquedas en profundidad y en anchura. Los mejores resultados obtenidos, utilizando un compilador Fortran, son del orden de los 11 minutos para el caso de mayor grado de especificación. No se analizó cómo se comporta la metodología de resolución en problemas de mayor escala. El éxito del algoritmo parece depender de la ubicación del óptimo global en el árbol de búsqueda.

De manera casi simultánea y con el objetivo de mejorar los tiempos de ejecución para instancias de mayor tamaño, Nguyen y Bagajewicz (2011b) presentaron una implementación de la búsqueda en árbol basada en conjuntos de corte (Gala y Bagajewicz, 2006b) en una arquitectura en paralelo. Se resolvieron problemas lineales de mediana y gran escala, de acuerdo a su propia clasificación, los cuales comprendían 24 y 52 variables, respectivamente. El algoritmo fue implementado y paralelizado en Fortran y ejecutado sobre una supercomputadora del *University of Oklahoma Supercomputing Center for Education and Research*, OSCER. Este equipo utilizaba CPU Intel Xeon con rangos de velocidad entre 2.0 to 2.4 GHz y contaba con 8768 GB RAM. Se garantizó la obtención de óptimos globales para los ejemplos mostrados. Los tiempos de cómputo reportados fueron del orden de 60 segundos para el caso mediano y de 10 hs para el segundo ejemplo. Este último tiempo fue considerado como aún aceptable por los propios autores.

Con posterioridad, Nguyen y Bagajewicz (2013) mejoran los tiempos de cómputo asociados a la resolución de los mismos problemas analizados en 2008 (Nguyen y Bagajewicz, 2008) a través de la propuesta de hibridizar la búsqueda en árbol con una heurística de búsqueda local. La idea básica es que las mejores soluciones encontradas en

algún punto del árbol combinatorio sean utilizadas como puntos iniciales para dicha heurística. La búsqueda local consta de dos etapas:

Etapas 1: su objetivo es encontrar una solución factible de mínimo costo a partir de propuestas de mediciones de alta calidad. El procedimiento consiste en:

- 1) Generar un vector  $s$  como la unión de las cinco mejores soluciones encontradas
- 2) Emplear una estrategia enumerativa para remover variables medidas de  $s$  para obtener una solución,  $mc$ , de menor costo.

Segunda etapa: su propósito es determinar si es posible mejorar a la solución  $mc$  reemplazando algunas mediciones que no pertenecen a ésta. Los pasos a seguir son:

- 3) Remover de  $mc$  un conjunto formado por  $Nr$  mediciones, obteniéndose el vector  $b$
- 4) Usar nuevamente una estrategia enumerativa para adicionar mediciones a  $b$  que no pertenezcan al vector  $mc$  de modo tal de minimizar el costo
- 5) Volver a 3)
- 6) Repetir hasta que todas las posibilidades de remoción de subconjuntos formados por  $Nr$  mediciones se hayan explorado.

El valor del parámetro  $Nr$  tiene impacto en el desempeño del algoritmo. Se observa que valores grandes contribuyen a verificar la optimalidad pero tienen un efecto negativo sobre los tiempos de cómputo. El método no garantiza encontrar el óptimo global y los tiempos reportados son del orden de 1 h 40 min para el caso de mayor complejidad, que corresponde al TEP con alto nivel de especificación.

Por su parte, Kotecha y col. (2007) utilizaron la técnica de Programación con Restricciones (*Constraint Programming*: CP) para minimizar la no observabilidad de las

fallas de un proceso en presencia de incertidumbre en la velocidad de falla de los sensores, en la probabilidad de ocurrencia de fallas, y en el modelo causa-efecto, con la finalidad de diseñar sistemas de instrumentación robustos y confiables. Estos autores concluyeron que la potencia de modelado de la CP permite reducir la dimensión del problema en comparación con las formulaciones MILP existentes en la literatura. En esta contribución también se presentaron casos de estudio de pequeña escala.

### **2.2.1.2. Técnicas de Programación Matemática.**

Todas las estrategias que forman parte de este grupo de metodologías se basan en considerar a las variables no medidas como mediciones con varianza infinita, lo que hace posible calcular de manera explícita algunas medidas de desempeño de la red en función de  $\mathbf{q}$ , y utilizar técnicas de programación matemática para resolver los PDRS así formulados. En el trabajo de Chmielewski y col. (1999) se presentó la idea de medir todas las variables, ya sea con instrumentos reales o ficticios de alto desvío estándar.

Posteriormente la propuesta de los mismos autores fue minimizar el costo de capital sujeto a restricciones sobre diferentes indicadores de desempeño, tales como la precisión de las estimaciones, la capacidad de detección de errores gruesos, la resiliencia de las estimaciones con respecto a los errores gruesos no detectados, y la probabilidad que la precisión de las estimaciones esté acotada para redes de sensores con probabilidades de falla fijas (Chmielewski, 2002). En la formulación propuesta la inversa de la matriz de covarianza es una función lineal de las variables de decisión binarias, sin embargo las restricciones son no lineales por lo que resulta un MINLP. El método usual de solución mediante la técnica B&B requiere encontrar en cada nodo el óptimo global del NLP resultante de la relajación de las restricciones enteras. Los autores modificaron la formulación previa convirtiendo las restricciones no lineales en restricciones de desigualdad convexas que tomaron la forma de desigualdades matriciales lineales, lo que

permitió obtener soluciones óptimas globales del PDRS mediante el algoritmo de búsqueda estándar de la técnica B&B. La reformulación se basó en la linealidad existente entre las variables de decisión binarias y la inversa de la matriz de covarianza asociada a las mediciones. Se resolvieron problemas de minimización sujetos a un conjunto de desigualdades matriciales lineales aplicando el programa MATLAB (*LMI Control Toolbox*), pero la dimensión de los ejemplos de aplicación presentados fue notoriamente baja: sólo cuatro corrientes de proceso.

Bagajewicz y Cabrera (2002) adoptaron el enfoque que utiliza sensores ficticios con alto desvío estándar para resolver los problemas originalmente definidos por el primer autor (Bagajewicz, 1997). Su contribución consistió en convertir las restricciones no lineales en lineales utilizando una transformación que linealiza el producto formado por una variable continua y una binaria al sustituirlo por un conjunto de desigualdades lineales. De esta manera transformaron el MINLP original en un MILP, que fue resuelto utilizando CPLEX del programa GAMS (Brooke y col., 1992). Utilizando la metodología propuesta resolvieron un problema de actualización de instrumentación con un número máximo de 18 variables binarias e indicaron que los problemas de mayor dimensión comenzaban a alcanzar las limitaciones de los algoritmos existentes.

Kelly y Zyngier (2008) propusieron una formulación MILP que permite considerar tres aspectos relacionados con un diseño óptimo de una RS: observabilidad, redundancia y precisión. En el planteo de la función objetivo, se considera el costo de instrumentación asociado a la medición primaria del caudal de la  $i$ -ésima corriente,  $c_{\alpha_i}$ , empleando una variable binaria  $\alpha_i$  tal que:  $\alpha_i=1$  si el caudal se mide, y  $\alpha_i=0$  en caso contrario. Por otra parte, se define otra variable binaria  $\beta_i$  tal que su valor es 1 (uno) si debe considerarse el costo de medir dicho caudal con más de un sensor, es decir, si existen requerimientos de

redundancia de hardware. De esta manera, las variables de optimización  $\alpha$  y  $\beta$  son binarias, y sus valores, 1 o 0, representan la colocación o no de un sensor en cada corriente y la indicación de redundancia de hardware o no, respectivamente.

Se incluyen asimismo costos de penalización o costos artificiales que aparecen cuando alguna de las restricciones no se verifican. Por ejemplo si el  $i$ -ésimo caudal no es observable, entonces la variable error  $eu_i$  asume un valor positivo y el costo total se incrementa de manera proporcional al producto de  $ru_i eu_i$ , donde  $ru_i$  es un parámetro de penalización que debe ser fijado. Por su parte, el costo de penalización de las restricciones de redundancia y precisión se representa mediante los productos  $(rv_i ev_i)$  y  $(rw_i ew_i)$ , respectivamente, en los cuales las variables involucradas tienen un significado análogo al de las restricciones de observabilidad. La función objetivo del PDRS planteado se define de la siguiente manera:

$$\text{Min}_{\alpha_i, \beta_i} \sum_{i=1}^I (c_{\alpha_i} \alpha_i + c_{\beta_i} \beta_i + ru_i eu_i + rv_i ev_i + rw_i ew_i) \quad (2.1)$$

siendo  $I$  el número total de corrientes del proceso.

En cuanto a las restricciones del problema de optimización, se utilizó la metodología de clasificación de variables desarrollada por Kelly en dos trabajos anteriores (Kelly, 1998 a, b) para expresar matemáticamente las condiciones que deben cumplirse para satisfacer los requerimientos de observabilidad, redundancia y precisión.

Si se analiza el conjunto de restricciones para observabilidad, puede apreciarse que el número de éstas es del orden de  $I \times I$ . Para modelarlas se incorporan variables de trabajo bidimensionales que asumen el valor cero si están asociadas a variables medidas y son libres en caso contrario. Además se fijan las tolerancias de observabilidad para cada

variable en cuestión. Lo mismo sucede para el conjunto de ecuaciones relativas a las restricciones de redundancia.

Para el caso de la precisión, el número de ecuaciones a generar es del orden de  $I \times J$ , siendo  $J$  el número de ecuaciones linealmente independientes asociadas a los balances de masa.

De lo expuesto anteriormente se deduce que el número total de restricciones a generar para explicitar el modelo en un formato orientado a ecuaciones, que sea resoluble mediante un método exacto, puede resultar excesivo para problemas de gran tamaño. Más aún, no sólo el número de variables binarias aumenta con el tamaño de las instancias a resolver sino también el número de variables de trabajo. Es bien conocido que esto origina tiempos de cómputo muy altos, o directamente la imposibilidad de resolver el problema por esta vía. Para salvar este inconveniente, los mismos autores sugirieron utilizar una heurística que descomponga el problema general de estimabilidad (observabilidad y redundancia) y precisión en dos subproblemas de la siguiente forma:

- 1) Resolver el problema MILP considerando sólo las restricciones de estimabilidad y retener el mayor conjunto de soluciones enteras factibles que se puedan alcanzar en un tiempo razonable.
- 2) Para cada solución hallada en el paso 1, resolver el problema LP de precisión asociado.
- 3) Si las soluciones para ambos problemas, estimabilidad y precisión, son aceptables desde el punto de vista del valor de las variables artificiales o variables error ( $eu$ ,  $ew$ ), entonces seleccionar aquella solución de costo mínimo
- 4) Si no, aplicar las estrategias de corte propuestas por Dogan y Grossmann (2006)



para remover del espacio de búsqueda del MILP todas o una porción de las soluciones factibles previamente obtenidas, y volver a resolver el MILP para satisfacer la estimabilidad de las variables.

Los ejemplos mostrados son pequeños. El primero corresponde a un proceso con 8 corrientes para el cual se resuelve el MILP completo. El segundo es un problema de actualización de instrumentación para un proceso de 63 corrientes, de las cuales 29 ya tienen sensores asignados. Para este caso, se propone determinar la localización de nuevos sensores de manera tal que con éstos y los existentes se satisfaga la observabilidad de todas las variables no medidas. Luego se asume que los instrumentos que garantizan la observabilidad se han instalado y se actualiza la RS para asegurar la redundancia de las variables medidas. Posteriormente se considera que se han ubicado los sensores que satisfacen la observabilidad y la redundancia, y se determina la localización de nuevos instrumentos de manera tal que se cumplan las restricciones de precisión.

Del análisis crítico del desempeño que exhiben los Algoritmos Exactos surge que su comportamiento es muy dependiente del problema de diseño particular que se trate y de su tamaño. En este sentido no tienen la robustez suficiente como para poder resolver una amplia gama de especificaciones ni la posibilidad de escalamiento. La desventaja más notoria es que la garantía de optimalidad, no puede ser alcanzada en muchos de los casos, dados los excesivos tiempos de cómputo que insumen.

Tanto para las técnicas basadas en búsqueda en árbol como en programación matemática, se advierte la tendencia a combinar metodologías de resolución exactas con técnicas heurísticas en el afán de brindar soluciones óptimas o cuasi óptimas para instancias de tamaño real. La naturaleza del PDRS Generales, en lo referente a la complejidad computacional que posee, hace que el tiempo de resolución asociado sea el

aspecto limitante a la hora de seleccionar la metodología de optimización más apropiada. Esta característica, compartida con una amplia gama de problemas de optimización combinatorios, ha impulsado el desarrollo de las llamadas técnicas metaheurísticas que proponen diferentes mecanismos de búsqueda de “buenas soluciones” en tiempos acotados.

### 2.2.2 Heurísticas

Dentro de estas técnicas de abordaje al PDRS pueden distinguirse, de manera muy general, tres grupos:

- a) Metodologías basadas en búsqueda local: los mecanismos de inspección de posibles soluciones óptimas están definidos sobre el entorno o vecindad de una solución inicial. Necesitan obviamente, de un punto inicial que puede ser hallado, por ejemplo, mediante la obtención de nodos de soluciones factibles conseguidas empleando MINLP.
- b) Metodologías de tipo constructivas: el ejemplo clásico de este grupo lo constituye el algoritmo goloso o voraz. Éste trata de construir una solución en diferentes etapas, incorporando en cada una de ellas un atributo, una variable en el caso del PDRS, que contribuya a optimizar el criterio de desempeño propuesto.
- c) Metodologías que proponen un conjunto (población) de soluciones que se modifican y mejoran de iteración en iteración utilizando operadores definidos previamente. En este grupo se incluyen los Algoritmos Evolutivos (AEs) en general, y como caso particular los Algoritmos Genéticos (AGs).

Inicialmente se resolvió el PDRS Mínimas, cuyas soluciones están formadas por el menor número de sensores que permite calcular todas las variables no medidas del

proceso. Esto implica que: el número de sensores de la red es un dato del problema de diseño, todas las variables no medidas son observables, y las mediciones son no redundantes. En el trabajo de Carnero y col. (2001a) se aplicó, en primer lugar, un algoritmo de tipo goloso para determinar la RS de mínimo costo. Para garantizar la obtención del óptimo global mediante dicho algoritmo es necesario que la función objetivo cumpla con ciertas características. Por lo tanto, se propuso la aplicación de un método basado en AEs, que permite tratar cualquier instancia de diseño, con el fin de abordar otros planteos del problema, con criterios de desempeño tanto monoobjetivo como multiobjetivo. A diferencia de un AG clásico, el procedimiento desarrollado emplea un mecanismo de sembrado de la población inicial utilizando una estrategia basada en álgebra lineal. Además los operadores genéticos clásicos se modificaron para evitar que se generen demasiadas soluciones no factibles.

También se estudió el PDRS Mínimas con objetivos múltiples. En tal sentido Carnero y col. (2001b) propusieron maximizar la confiabilidad de la RS y minimizar el costo de instrumentación sujeto a restricciones de observabilidad. Empleando el enfoque de Srinivas y Deb (1994), se utilizó el concepto de dominancia de Pareto para la asignación de medidas de bondad a las soluciones. La metodología asigna rango uno a todos los individuos (soluciones) no dominados y los elimina temporalmente de la población. Este conjunto constituye el primer frente. A los siguientes individuos no dominados se les asigna rango dos, y así sucesivamente, de modo tal que al final de este procedimiento, la población queda dividida en un cierto número de frentes. Además se incorpora una fase de construcción y mantenimiento de un conjunto de élite que evita la pérdida de buenas soluciones, y una técnica destinada a compartir soluciones con el fin de mantener diversidad en la población.

Por su parte Gerkens y Heyen (2005) resolvieron el PDRS Redundantes para dos procesos, que involucran 212 y 618 variables respectivamente. Las RS Redundantes comprenden una cantidad de instrumentos superior a la necesaria para conseguir una estimación única de todas las variables. Utilizaron dos técnicas de paralelización de un AG clásico que les permitieron reducir los tiempos de cómputo al orden de 10 min para el primer ejemplo y 4 hs para el segundo. En este trabajo se reporta la resolución de los casos de estudio más grandes, publicados en la literatura, en tiempos de cómputo aceptables. Nada puede decirse de la calidad de las soluciones encontradas ya que no se disponen de límites inferiores para contrastar. Dada la naturaleza probabilística de la heurística propuesta deberían haberse mostrado varias ejecuciones de la misma para poder analizar su desempeño.

En cuanto al PDRS Generales, Carnero y col. (2004) presentaron un enfoque híbrido para determinar la RS que minimiza el costo de instrumentación en el ciclo de vida de la planta y satisface requerimientos sobre la precisión y disponibilidad de las estimaciones de un conjunto de variables claves. Dado que la incorporación de conocimiento específico del problema mejora drásticamente el desempeño de un AE, los autores desarrollaron un procedimiento *ad hoc* de generación de la población inicial de tal manera de asegurar la factibilidad de todas las soluciones con respecto a la estimabilidad de las variables claves del proceso. Una variable es estimable si es medida o, no medida pero observable. También se realizó la incorporación de información del problema en otras etapas de la evolución. En tal sentido, se propuso un AE, basado en AGs, con población estructurada en la forma de vecindades, cuyo desempeño se mejoró empleando una estrategia de búsqueda local. El algoritmo propuesto fue empleado para la resolución del Problema (1.3) y los resultados alcanzados mostraron que el mismo logra

un buen balance entre la exploración y la explotación del espacio de búsqueda (Carnero y col., 2005).

Posteriormente Alsogaray y col. (2005) formularon un PDRS Generales multiobjetivo. Consideraron como funciones objetivo en conflicto al costo de adquisición de instrumentos y la precisión de la estimación de un conjunto de variables claves. Resolvieron el problema empleando los métodos: *Non-dominated Sorting Genetic Algorithm – II* (Deb y col., 2002) y *Strength Pareto Evolutionary Algorithm II* (Zitzler, 1999, 2001), que corresponden a la segunda generación de AEs multiobjetivo. En este trabajo se implementó un mecanismo de reparación de soluciones no factibles originadas por la aplicación de los operadores genéticos. Este consiste en determinar el conjunto de variables claves que no son estimables para dicha solución e incorporarlas, de a una por vez al conjunto de mediciones, hasta conseguir que se satisfaga la estimabilidad de todas las variables claves.

Por su parte, Brewer y col. (2007) resolvieron el PDRS para observabilidad en sistemas dinámicos no lineales donde se tienen en cuenta las interacciones entre los sensores individuales utilizando análisis de componentes principales. La observabilidad, definida como la propiedad de un sistema que permite la reconstrucción del estado de las variables dado un conjunto de datos proporcionado por las salidas del sistema, fue descrita mediante la matriz Gramiano de Observabilidad. Ésta puede interpretarse como la energía generada por las salidas del sistema (la respuesta de los sensores) para las perturbaciones en los estados iniciales. Se definieron medidas sobre dicha matriz, tales como el mínimo valor propio, la inversa de la traza o el determinante para establecer el grado o nivel de observabilidad asociado a diferentes configuraciones de sensores. El objetivo del problema propuesto fue diseñar RS que maximicen la información que puede extraerse acerca de la operación de la planta. Se propusieron dos metodologías de

resolución, la primera consistió en la localización secuencial de un sensor a la vez, con una heurística del tipo voraz. El algoritmo converge rápidamente a una solución y su implementación es sencilla. La segunda estrategia considera la localización simultánea de un conjunto de sensores para lo cual se consideró un AG de tipo estándar con una función de evaluación que tiene en cuenta una medida de la información global suministrada por la solución corriente y el costo asociado a esa estructura de sensores.

### 2.3 Conclusiones

Si el PDRS Generales se formula como un MINLP, existen dos dificultades principales. La primera está relacionada con la naturaleza combinatorial del mismo y la segunda con la complejidad de las relaciones funcionales entre las variables.

A medida que el número de variables binarias crece, el tiempo de ejecución del algoritmo se incrementa exponencialmente, ya que del análisis de complejidad computacional surge que los PDRS formulados como MINLP se caracterizan por ser NP-completos (Floudas, 1995), esto es, no se conocen algoritmos polinomiales para encontrar su solución, por lo tanto sólo pueden resolverse instancias de tamaño pequeño. Al mismo tiempo la no linealidad de estos problemas puede ser no convexa, lo que dificulta la garantía de optimalidad.

Para instancias del problema de tamaño pequeño la implementación de algoritmos de búsqueda en árbol resulta en la obtención del óptimo global. Dichas técnicas sólo requieren que el conjunto solución en cada nodo pueda particionarse en dos subconjuntos mutuamente excluyentes, y disponer de un algoritmo para el cálculo de un límite inferior de la función objetivo para cualquier solución en un conjunto dado. Por lo tanto, de verificarse estas condiciones, es una forma de solución aplicable a cualquier tipo de

función objetivo y conjunto de restricciones. Por otra parte, si el problema admite formulaciones tales como un MINLP o un MILP, entonces, una técnica del tipo B&B con relajación de las variables binarias en cada nodo puede utilizarse con éxito.

Cuando el número de posibles ubicaciones de los instrumentos aumenta, el costo computacional asociado a todas las metodologías del tipo B&B se incrementa de tal forma que se vuelven impracticables. En estos casos la utilización de estrategias basadas en heurísticas aparece como una alternativa válida, ya que si bien carecen de garantías formales sobre su desempeño, son útiles para hallar “buenas” soluciones a problemas que representan escenarios más realistas. Una de las heurísticas probadas en la resolución del PDRS han sido los AGs. En líneas generales puede decirse que el comportamiento de un AG estándar es pobre, razón por la cual, en la mayoría de los trabajos reportados para la resolución del PDRS, se muestran AGs que incluyen alguna clase de estrategia de enriquecimiento que les permitan encontrar mejores soluciones.

Uno de los principales inconvenientes de los AGs es la pérdida de buen material genotípico en el proceso de evolución. Traducido al PDRS, es la pérdida de asociaciones de variables medidas que se repiten en las soluciones de alta calidad, y que pueden comenzar a formarse en etapas tempranas del algoritmo, para luego desaparecer merced al operador de cruzamiento.

Esta desventaja de los AGs ha llevado en los últimos años al desarrollo de otros AEs tales como los Algoritmos de Estimación de Distribuciones, AEDs. Estos aparecen como estrategias superadoras respecto de los AGs clásicos, lo cual se ve reflejado en la creciente cantidad de artículos donde se muestran las capacidades de los AEDs para la resolución de problemas combinatorios complejos.

Del análisis del estado del arte se desprende que los esfuerzos deberían estar dirigidos a poder resolver problemas de formulaciones más complejas, de instancias cada vez más grandes y donde se pueda garantizar la convergencia en tiempos aceptables a un conjunto de buenas soluciones (si no óptimas). En este sentido, las metodologías de resolución, tales como las metaheurísticas, que incluyen más de un método de optimización, resultan las más atractivas dada su flexibilidad y robustez.

## 2.4 Nomenclatura

<b>b</b>	Vector resultante al remover mediciones de <b>mc</b>
<b>c</b>	Vector de costos de instrumentación
$c_{\alpha_i}$	Costo de medición primaria de la variable $i$
$c_{\beta_i}$	Costo de medición secundaria de la variable $i$
$eu$	Variable error asociada a la restricción de observabilidad
$ev$	Variable error asociada a la restricción de redundancia
$ew$	Variable error asociada a la restricción de precisión
$I$	Número de variables de proceso
$J$	Número de ecuaciones linealmente independientes
<b>mc</b>	Vector solución obtenido a partir de <b>s</b>
$Nr$	Número de mediciones a remover
<b>q</b>	Vector de variables binarias



---

$ru$	Parámetro de penalización asociado a restricción de observabilidad
$rv$	Parámetro de penalización asociado a restricción de redundancia
$rw$	Parámetro de penalización asociado a restricción de precisión
$\mathcal{S}_\sigma$	Conjunto de variables de proceso sujetas a restricciones de precisión
$s$	Vector resultante de la unión de soluciones

#### Letras Griegas

$\alpha_i$	Variable binaria asociada a la medición primaria de la corriente $i$
$\beta_i$	Variable binaria asociada a la medición redundante de la corriente $i$
$\hat{\sigma}_{lp}$	Desvío estándar de la estimación de la $lp$ -ésima variable
$\sigma_{lp}^*$	Desvío estándar superior de la estimación de la $lp$ -ésima variable

#### 2.5 Acrónimos

AE	Algoritmo Evolutivo
AED	Algoritmo de Estimación de Distribuciones
AG	Algoritmo Genético
B&B	Algoritmo de <i>Branch and Bound</i>
CP	<i>Constraint Programming</i>
CSRT	<i>Continuous Stirred Tank Reactor</i>

---

DRS	Diseño de Redes de Sensores
LP	<i>Linear Programming</i>
MFP	<i>Mineral Flotation Process</i>
MINLP	<i>Mixed Integer Non Linear Programming</i>
NLP	<i>Non Linear Programming</i>
OSKER	<i>Oklahoma Supercomputing Center for Education and Research</i>
PDRS	Problema de Diseño de Redes de Sensores
RS	Red de Sensores
TEP	<i>Tennessee Eastman Process</i>

## 2.6 Referencias

- Alsogaray, M., Hernández, J., Carnero, M., Sánchez, M. Performance Comparison of Multiobjective Evolutionary Algorithms for Optimal Sensor Network Design. *Proceeding of ENPROMER 2005*, Río de Janeiro Disponible en CD, (2005).
- Bagajewicz, M., E. Cabrera. A New MILP Formulation for Instrumentation Network Design and Upgrade. *AIChE J.* **48**, 10, 2271-2282, (2002).
- Bagajewicz, M. Design and Retrofit of Sensor Networks in Process Plants. *AIChE J.*, **43**, 2300-2306, (1997).
- Bhushan, M., Rengaswamy, R. Design of Sensor Network Based on the Signed Directed Graph of the Process for Efficient Fault Diagnosis. *Ind. Eng. Chem. Res.*, **39**, 999–1019, (2000).

- Brewer, J., Zuyi Huang, Z., Singh, A., Misra, M., Hahn, J. Sensor Network Design via Observability Analysis and Principal Component Analysis. *Ind. Eng. Chem. Res.*, **46** (24), 8026–8032, (2007).
- Brooke, A.D., Kendrick, D., Meeraus, A. GAMS: A User's Guide. *Scientific Press*, San Francisco, (1992).
- Carnero, M., Hernández, J., Sánchez, M., Bandoni, A. An evolutionary approach for the design of non-redundant sensor networks. *Ind. Eng. Chem. Res.*; **40**, 5578-84, (2001a).
- Carnero, M., Hernández, J., Sánchez, M; Bandoni, A. Multiobjective Evolutionary Optimization in Sensor Network Design. *Proceedings of ENPROMER 2001*, **1**, 325-330. Santa Fe, (2001b).
- Carnero, M., Hernández, J., Sánchez, M., Bandoni, A. Availability of Key Process Variable Estimates: An Evolutionary Approach. *Escape 14*, Lisboa, Portugal, (2004).
- Carnero, M., Hernández, J., Sánchez, M., Bandoni, A. On the Solution of the Instrumentation Selection Problem. *Ind. Eng. Chem. Res.*, **44**, 2, 358-367, (2005).
- Chmielewski, D., Palmer, T., Manousiouthakis, V. On the Theory of Optimal Sensor Placement, *AIChE J.*, **48**, 1001–1012, (2002).
- Chmielewski, D., Palmer, T., Manousiouthakis, V. Cost Optimal Retrofit of Sensor Networks with Loss Estimation Accuracy Bounds. *AIChE Annual Meeting*, Dallas, TX, (1999).
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6** (2), 182-197, (2002).
- Dogan, M.E., Grossmann, I.E. A decomposition method for the simultaneous planning and scheduling of single-stage continuous multiproduct plants. *Ind Eng Chem Res.*, **45**, 299–315, (2006).

- Floudas, C. *Nonlinear and Mixed-Integer Optimization – Fundamentals and Applications*, Oxford University Press, New York, 1995.
- Gala, M., Bagajewicz, M. Rigorous Methodology for the Design and Upgrade of Sensor Networks using Cutsets *Industrial and Engineering Chemistry Research*. **45**, N 21, 6679-6686, (2006a).
- Gala, M., Bagajewicz, M. Efficient Procedure for the Design and Upgrade of Sensor Networks using Cutsets and Rigorous Decomposition. *Industrial and Engineering Chemistry Research*, **45**, 21, 6687-6697, (2006b).
- Gerkens, C., Heyen, G. Use of parallel computers in rational design of redundant sensor networks. *Comput. Chem. Eng.* **29**, 1379-1387, (2005)
- Kelly JD. A regularization approach to the reconciliation of constrained data sets. *Comp. Chem. Eng.*, **22**, 1771–1788, (1998a).
- Kelly JD. On finding the matrix projection in the data reconciliation solution. *Comp. Chem. Eng.*, **22**, 1553–1557, (1998b).
- Kelly, J., Zyngier, D. A new and improved MILP formulation to optimize observability, redundancy and precision for sensor network problems. *AIChE J.*, Vol. 54, pp: 1282–1291, (2008).
- Kotecha PR, Bhushan M, Gudi RD. Constrained programming based robust sensor network design. *Ind. Eng. Chem. Res.*; **46**, 5985-99, (2007).
- Luong, M., Maquin, D., Huynh, C., Ragot, J. Observability, Redundancy, Reliability and Integrated Design of Measurement Systems, *Proc. of 2nd. IFAC Symposium on Intelligent Components and Instrument Control Applications*, Budapest, Hungría, (1994).
- Madron, F., Veverka, V. Optimal Selection of Measuring Points in Complex Plants by Linear Models. *AIChE J.*, **38**, 227–236, (1992).
- Mah, R.S.H. *Chemical Process Structures and Information Flows*, Butterworths, Boston, (1990).

- Mah, R.S.H., Stanley, G. Dowing, D. Reconciliation and Rectification of Process Flow and Inventory Data. *Ind. Eng. Chem. Process Des. Dev.*, **15**, 175-183, (1976).
- Meyer, M., Le Lann, J.M., Koehret, B., Enjalbert, M. Optimal Location of Sensor Location on a Complex Plant Using a Graph Oriented Approach. *Computers and Chemical Engineering*, **18** Supp., S535-S540, (1994).
- Narasimhan, S., Jordache, C. *Data Reconciliation and Gross Error Detection*, Gulf Publishing Company, Houston, (2000).
- Nguyen, D.Q., Bagajewicz, M. Design of Nonlinear Sensor Networks for Process Plants. *Ind. Eng. Chem. Res.*, **47**, 5529–5542, (2008).
- Nguyen, D.Q., Bagajewicz, M. New efficient breath-first/level traversal methodology for the Design and Upgrade of Sensor Networks. *AIChE J*, **57**, 5, 1302-1309, (2011a).
- Nguyen, D.Q., Bagajewicz, M. Parallel Computing Approaches to Sensor Network Design Using the Value Paradigm. *Computers & Chemical Engineering*, **35** (6), 1119-1134, (2011b).
- Nguyen, D.Q., Bagajewicz, M. Efficient Approximate Methods for the Design and Upgrade of Sensor Networks. *Industrial and Engineering Chemistry Research*, **52**(1), 83-90, (2013).
- Ricker, N.L., Lee, J.H. Nonlinear Modeling and State Estimation for the Tennessee Eastman Challenge Process. *Computers & Chemical Engineering*, **19** (9), 983–1005, (1995).
- Romagnoli, J., Sánchez, M. *Data processing and reconciliation in chemical process operations*. Academic Press, San Diego, (2000).
- Smith, H.W., Ichiyen, N. Computer Adjustment of Metallurgical Balances. *Canadian Mining and Metallurgical Bulletin*, **66** (737), 97-100, (1973)
- Srinivas, N., Deb, K. Multiobjective Optimization Using Nondominated Sorting In Genetic Algorithms. *Journal of Evolutionary Computation*, **2**, 221-248, (1994).

- Vaclavek, V., Loucka, M. Selection of Measurements Necessary to Achieve Multicomponent Mass Balances in Chemical Plant. *Chem. Engng. Sci.*, **31**, 1199-1205, (1976).
- Zitzler, E., Thiele, L. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, **3**(4), 257-272. (1999).
- Zitzler, E., Laumanns, M., Thiele, L. SPEAII: Improving the strength pareto evolutionary algorithm. *Technical Report 103*, Swiss Federal Institute of Technology, Zurich, Switzerland, (2001).

# ***Capítulo 3***

**Algoritmos con Búsquedas por  
Entorno**

### 3.1 Introducción

En este capítulo se aborda la solución del PDRS de costo mínimo sujeto a restricciones de estimabilidad y precisión sobre un conjunto de variables claves mediante el empleo de metaheurísticas de búsqueda local y de tipo híbrido.

El PDRS es un problema de optimización combinatoria complejo cuya resolución exacta sólo puede lograrse para instancias relativamente pequeñas. Si el objetivo es encontrar soluciones a problemas de mayor envergadura, entonces las heurísticas aparecen como una alternativa válida. Dentro de éstas, las técnicas de búsqueda local han sido utilizadas en una amplia variedad de problemas de optimización combinatoria.

En años recientes una metodología de búsqueda local que ha recibido considerable atención es la metaheurística denominada Búsqueda Tabú (BT). Ésta ha sido aplicada con éxito a la solución de problemas de optimización dentro del área de ingeniería química (Lin y Miller, 2004 a - b; Cavin y col., 2004) lo cual ha motivado el desarrollo de heurísticas dentro de este paradigma que se apliquen a la resolución del PDRS. La técnica guía un procedimiento de búsqueda local que permite explorar el espacio de soluciones más allá de un óptimo local. La búsqueda local utiliza un operador llamado movimiento, que define una vecindad de una solución dada, junto con una memoria adaptiva del proceso, que hace al método más flexible y eficaz.

Otra metodología de resolución que utiliza estrategias de mejora de soluciones a través de búsqueda por entornos es la llamada Búsqueda Dispersa, BD. A diferencia de la BT, la BD trabaja con un conjunto de soluciones que evolucionan mediante dos mecanismos fundamentales: la combinación de dos o más soluciones, y la mejora de la solución resultante mediante diferentes técnicas de búsqueda local. Ha sido aplicada a distintos problemas de optimización discretos de reconocida complejidad computacional



tales como: Problema de Entrenamiento de Redes Neuronales, Problemas de Ruteo, Coloreo de Grafos, Planeamiento de Tareas, etc, (Laguna y col., 2003, Martí y col., 2006).

A continuación se formula el PDRS que se aborda en esta tesis, y se presentan diferentes estrategias de resolución. Inicialmente se proponen técnicas de búsqueda local basadas en BT, luego se describe un método híbrido que utiliza BD en combinación con BT.

### 3.2 Formulación del problema

El PDRS de mínimo costo que satisface las restricciones de estimabilidad y precisión sobre un determinado conjunto de variables claves se formula como sigue:

$$\begin{aligned}
 & \text{Min} && \mathbf{c}^T \mathbf{q} \\
 & \text{s.t.} && \\
 & && \hat{\sigma}_{lp}(\mathbf{q}) \leq \sigma_{lp}^* \quad \forall lp \in \mathcal{S}_\sigma \\
 & && E_{le}(\mathbf{q}) \geq 1 \quad \forall le \in \mathcal{S}_\varepsilon \\
 & && \mathbf{q} \in \{0,1\}^I
 \end{aligned} \tag{3.1}$$

siendo  $\mathbf{c}$  el vector de costos de adquisición de los instrumentos disponibles,  $\mathbf{q}$  el vector de variables binarias tal que,  $q_i = 1$  si el caudal másico  $z_i$  es medido y  $q_i = 0$  en caso contrario,  $E_{le}$  el Grado de Estimabilidad de la variable  $le$  (Bagajewicz y Sánchez, 1999),  $\hat{\sigma}_{lp}$  el Desvío Estándar de la Estimación de la variable  $lp$  obtenida después de aplicar un procedimiento de Reconciliación de Datos y  $\sigma_{lp}^*$  su correspondiente límite superior.

Las restricciones de estimabilidad y precisión se imponen sobre las variables pertenecientes a los subconjuntos  $\mathcal{S}_\varepsilon$  y  $\mathcal{S}_\sigma$ , siendo  $\mathcal{S}_\sigma$  un subconjunto de  $\mathcal{S}_\varepsilon$ . Se

asume que hay disponible un solo sensor para medir cada variable del proceso y que no existen restricciones para la localización de los mismos.

### 3.2.1 Restricciones de Estimabilidad

Si se denota  $\mathcal{A}(i)$  al conjunto de todas las posibles combinaciones de  $i$  mediciones y  $\mathcal{A}(l)$ , al  $l$ -ésimo elemento (combinación) de este conjunto; la definición de Grado de Estimabilidad se enuncia a continuación: “una variable  $i$  (medida o no) tiene Grado de Estimabilidad  $E_i$  si: resulta estimable después de la eliminación de cualquier combinación  $\mathcal{A}(E_i-1) \in \mathcal{A}(E_i-1)$  y, resulta no observable cuando al menos un conjunto  $\mathcal{A}(E_i) \in \mathcal{A}(E_i)$  se elimina.

Del análisis de la definición anterior surge que, las restricciones impuestas sobre la estimabilidad de las variables claves se satisfacen si éstas son medidas u observables. En consecuencia el análisis de la factibilidad de una solución  $\mathbf{q}$  con respecto a la restricción de estimabilidad se reduce a determinar si las variables claves no medidas contenidas en  $\mathbf{q}$  son observables. Para ello se eligió utilizar un procedimiento de Clasificación de Variables flexible, tal como el basado en factorizaciones ortogonales Q-R (Sánchez y Romagnoli, 1996).

Los procedimientos matemáticos de Clasificación de Variables permiten categorizar las variables medidas y no medidas de un proceso teniendo como datos el modelo que representa su operación y un conjunto de instrumentos. Dado un proceso cuya operación en estado estacionario se representa mediante un conjunto de ecuaciones algebraicas no lineales, la linealización de las ecuaciones del modelo alrededor de un punto operativo de estado estacionario se representa mediante el siguiente conjunto de ecuaciones lineales

$$\mathbf{D} \mathbf{z} = \mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u} = \mathbf{d} \quad (3.2)$$

donde  $\mathbf{z}$  es el vector de variables del proceso de dimensión  $I$ ,  $\mathbf{x}$  representa el vector de variables medidas de dimensión  $I_x$ ,  $\mathbf{u}$  corresponde al vector de variables no medidas de dimensión  $I_u$ ,  $\mathbf{A}$  ( $J \times I_x$ ) es el jacobiano de las ecuaciones del modelo respecto de las variables medidas,  $\mathbf{B}$  ( $J \times I_u$ ) representa el jacobiano de las ecuaciones del modelo respecto de las variables no medidas,  $\mathbf{D}$  es una matriz que comprende a las submatrices  $\mathbf{A}$  y  $\mathbf{B}$  de dimensión compatible, y  $\mathbf{d}$  corresponde a un vector de constantes de dimensión  $J$ .

La descomposición Q-R de la matriz  $\mathbf{B}$  permite obtener las matrices  $\mathbf{Q}_u$ ,  $\mathbf{R}_u$  y la matriz de permutación  $\mathbf{\Pi}_u$ , tal que

$$\mathbf{B} \mathbf{\Pi}_u = \mathbf{Q}_u \mathbf{R}_u, \quad (3.3)$$

a su vez  $\mathbf{Q}_u$  y  $\mathbf{R}_u$  pueden dividirse en

$$\mathbf{Q}_u = [\mathbf{Q}_{u1} \quad \mathbf{Q}_{u2}] \quad \mathbf{R}_u = \begin{bmatrix} \mathbf{R}_{u1} & \mathbf{R}_{u2} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (3.4)$$

con  $r_u = \text{rango}(\mathbf{B}) = \text{rango}(\mathbf{R}_{u1})$ . Cabe destacar que  $\mathbf{Q}_u$  es una matriz ortogonal y  $\mathbf{R}_{u1}$  es una matriz triangular superior de dimensión  $r_u$ . De la misma manera las variables no medidas pueden separarse en dos conjuntos

$$\mathbf{\Pi}_u^T \mathbf{u} = \begin{bmatrix} \mathbf{u}_{r_u} \\ \mathbf{u}_{I_u-r_u} \end{bmatrix} \quad (3.5)$$

Premultiplicando el sistema de ecuaciones generales  $\mathbf{D} \cdot \mathbf{z} = \mathbf{d}$  por  $\mathbf{Q}_u^T = \mathbf{Q}_u^{-1}$  se obtiene

$$\begin{bmatrix} \mathbf{Q}_1^T \mathbf{A} & \mathbf{R}_1 & \mathbf{R}_2 \\ \mathbf{Q}_2^T \mathbf{A} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u}_r \\ \mathbf{u}_{I_u-r} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_1^T \\ \mathbf{Q}_2^T \end{bmatrix} \mathbf{d} \quad (3.6)$$

Las variables contenidas en el vector  $\mathbf{u}_{r_u}$  de dimensión  $r_u$  pueden expresarse en función de las demás variables de la siguiente manera

$$\mathbf{u}_{r_u} = \mathbf{R}_1^{-1} \mathbf{Q}_1^T \mathbf{d} - \mathbf{R}_1^{-1} \mathbf{Q}_1^T \mathbf{A} \mathbf{x} - \mathbf{R}_1^{-1} \mathbf{R}_2 \mathbf{u}_{I_u - r_u} \quad (3.7)$$

Las siguientes conclusiones surgen del análisis de las ecuaciones anteriores:

➤ Conclusión 1

Si  $\text{rango}(\mathbf{R}_u) = r_u = I_u$ , entonces todas las variables no medidas son estimables con las mediciones existentes

Si  $\text{rango}(\mathbf{R}_u) = r_u < I_u$ , entonces al menos  $(I_u - r_u)$  variables no pueden calcularse.

➤ Conclusión 2

Las variables en el subconjunto  $\mathbf{u}_{I_u - r_u}$  representan la mínima cantidad y la localización de las mediciones necesarias para satisfacer la condición de estimabilidad completa del sistema, es decir que todas las variables no medidas sean observables.

➤ Conclusión 3

La matriz  $\mathbf{Q}_{u2}^T$  obtenida de la factorización Q-R ( $\mathbf{B}$ ) es tal que

$$\mathbf{Q}_{u2}^T \mathbf{B} = \mathbf{0} \quad (3.8)$$

La factorización Q-R de  $\mathbf{B}$  da información sobre la factibilidad de calcular las variables no medidas. Las variables contenidas en  $\mathbf{u}_{I_u - r_u}$  son no observables. Con respecto a las incluidas en  $\mathbf{u}_{r_u}$ , algunas pueden calcularse directamente utilizando las mediciones

disponibles, mientras que otras dependen de los valores de las variables presentes en  $\mathbf{u}_{1-r_u}$ . Para clasificar las variables en  $\mathbf{u}_u$  es necesario inspeccionar las filas de la matriz

$$\mathbf{R}_{1U} = \mathbf{R}_{u1}^{-1} \mathbf{R}_{u2} \quad (3.9)$$

Una variable en  $\mathbf{u}_u$  es observable si la correspondiente fila en la matriz  $\mathbf{R}_{1U}$  es el vector nulo, en caso contrario es no observable.

La clasificación de las variables medidas resulta de examinar la matriz  $\mathbf{G}_x$

$$\mathbf{G}_x = \mathbf{Q}_{u2}^T \mathbf{A} \quad (3.10)$$

Las columnas de  $\mathbf{G}_x$  que son vectores nulos representan las variables medidas que no participan en ecuaciones con otras mediciones, por lo tanto corresponden a mediciones no redundantes. Las columnas restantes de  $\mathbf{G}_x$  están asociadas a las mediciones redundantes.

### 3.2.2 Restricciones de Precisión

La precisión de las estimaciones de las variables medidas contenidas en  $\mathbf{q}$  se calcula, luego de aplicar el procedimiento de Reconciliación de Datos, como la raíz cuadrada de los elementos de la diagonal de la matriz  $\mathbf{S}_x$ ,

$$\mathbf{S}_x = \text{Cov}(\hat{\mathbf{x}}) = (\mathbf{I}_x - \mathbf{M}_x) \mathbf{S}_y (\mathbf{I}_x - \mathbf{M}_x)^T \quad (3.11)$$

siendo  $\hat{\mathbf{x}}$  el valor corregido de las variables medidas,  $\mathbf{I}_x$  la matriz identidad de dimensión igual al número de variables medidas,  $\mathbf{M}_x = \mathbf{S}_y \mathbf{G}_x^T (\mathbf{G}_x \mathbf{S}_y \mathbf{G}_x^T)^{-1} \mathbf{G}_x$  y,  $\mathbf{S}_y$  representa la matriz de covarianza de los errores aleatorios de las mediciones,  $\boldsymbol{\varepsilon}$ , que se suponen tienen una distribución normal con media nula.

De forma análoga, las precisiones de las variables no medidas observables se calculan como la raíz cuadrada de los elementos de la diagonal de la matriz  $\mathbf{S}_{u0}$ ,

$$\mathbf{S}_{\mathbf{u}_0} = \text{Cov}(\hat{\mathbf{u}}_0) = \mathbf{H}_{\mathbf{u}_0} \mathbf{S}_x \mathbf{H}_{\mathbf{u}_0}^T, \quad (3.12)$$

siendo  $\hat{\mathbf{u}}_0$  el vector de estimaciones de las variables no medidas observables, y  $\mathbf{H}_{\mathbf{u}_0}$  la matriz formada por las filas de  $\mathbf{H}_u = -\mathbf{R}_{\mathbf{u}}^{-1} \mathbf{Q}_{\mathbf{u}}^T \mathbf{A}$  tales que las mismas filas de la matriz de inspección  $\mathbf{R}_{\mathbf{u}}$  son vectores nulos.

### 3.3 Nuevas Estrategias de Diseño de Redes de Sensores basadas en Búsqueda Tabú

La BT es una técnica metaheurística que utiliza un procedimiento de búsqueda local para explorar el espacio de soluciones, junto con mecanismos de memoria adaptiva diseñados para evitar el estancamiento en mínimos locales y la visita cíclica de las mismas soluciones. La información histórica acerca del proceso de búsqueda de soluciones se almacena en las llamadas Listas Tabú.

A partir de una solución inicial  $\mathbf{q}_0$ , se define en cada iteración una vecindad de posibles soluciones  $\mathcal{N}(\mathbf{q})$ , obtenida modificando la solución de la iteración previa  $\mathbf{q}$  mediante una secuencia de movimientos. Los elementos de dicha vecindad se examinan para determinar el mejor de ellos que está ausente de la lista tabú,  $\mathbf{q}'$ . Esta solución se selecciona como punto de partida para una nueva iteración, aun cuando no mejore la solución anterior  $\mathbf{q}$ , a la vez que se guarda la mejor solución encontrada hasta el momento, representada como  $\mathbf{q}^*$ .

El método lleva un registro de los movimientos efectuados para alcanzar  $\mathbf{q}'$  desde  $\mathbf{q}$  en cada iteración. Esta memoria del proceso de búsqueda determina qué soluciones pueden ser alcanzadas por un movimiento a partir de la solución actual. Normalmente esto se lleva a cabo a través de una lista de movimientos temporalmente prohibidos

llamada Lista Tabú basada en lo Reciente (LTR), la cual se actualiza en cada iteración. Pasado un cierto número de iteraciones, llamado período tabú ( $pt$ ), el movimiento se permite nuevamente. La LTR representa la memoria a corto plazo del proceso. El estado tabú asociado a un movimiento puede eventualmente ser revocado si se satisface algún criterio de aspiración como por ejemplo: la generación de una solución de mejor calidad que cualquier otra visitada previamente durante la búsqueda

En contraste, la Lista Tabú basada en la Frecuencia (LTF) almacena información acerca de la periodicidad con que ciertos movimientos se han realizado. Dicha lista representa la memoria a largo plazo de la técnica, y permite la diversificación de la búsqueda al seleccionar movimientos que han sido menos utilizados o nunca se han elegido.

A continuación se describen los aspectos fundamentales del algoritmo propuesto (Carnero y col., 2005 a - b) para resolver el PDRS planteado en la sección anterior utilizando la Búsqueda Tabú Clásica (BTC).

### **3.3.1 Algoritmo basado en Búsqueda Tabú Clásica**

#### **3.3.1.1. Generación de la Solución Inicial**

La generación de la solución inicial de la BT puede realizarse al azar, o mediante algún mecanismo que permita incluir el conocimiento específico del problema con la finalidad de partir de una solución de mejor calidad. En este trabajo se opta por la segunda alternativa. Se propone generar una población de soluciones iniciales dentro del espacio de soluciones factibles desde el punto de vista de la estimabilidad (Carnero y col., 2005c) y seleccionar la mejor solución encontrada como solución inicial de la BT. Para ello se desarrolló la técnica que se describe a continuación.

Cada solución de la población generada se representa por medio de una matriz  $\mathbf{T}$  de dimensión  $[nr \times (J+1)]$ , donde  $nr$  es el número de variables requeridas y  $J$  indica el número de filas de la matriz  $\mathbf{D}$ .

Las primeras  $J$  columnas de  $\mathbf{T}$  contienen la posición ordinal en  $\mathbf{D}$  de la ecuación o combinaciones de ecuaciones que se utilizan para el cálculo de cada variable requerida. En caso de ser necesario, se agregan ceros hasta completar las  $J$  columnas. La última columna de  $\mathbf{T}$  corresponde al Índice de Medición,  $MI$ , que es una variable aleatoria discreta asociada a cada variable requerida. Los valores asumidos para  $MI$  pueden ser 1, si corresponde a una variable medida no redundante, 0 si es una variable no medida y -1 si corresponde a una variable medida redundante.

Se adopta una convención especial para escribir la posición ordinal en  $\mathbf{D}$  correspondiente a las ecuaciones involucradas en el cálculo de una variable requerida. Para cada fila de  $\mathbf{T}$ , el primer lugar se ocupa con la posición ordinal de una ecuación con coeficiente distinto de cero. Los otros lugares se completan con las posiciones ordinales de las ecuaciones con coeficientes cero y los restantes lugares con ceros.

Para ilustrar la representación de una solución, considérese que las variables  $z_2$  y  $z_4$  son las variables requeridas de un proceso cuya operación se representa mediante la siguiente matriz  $\mathbf{D}$ ,

$$\mathbf{D} = \begin{bmatrix} 1 & -1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 \end{bmatrix}$$

La variable  $z_2$  aparece con coeficientes distintos de cero en las filas 1 y 2. Por lo tanto  $z_2$  puede ser calculada de cuatro maneras diferentes: a) utilizando la ecuación 1; b)



utilizando la ecuación 2; c) utilizando una combinación de las ecuaciones 1 y 3; d) utilizando una combinación de las ecuaciones 2 y 3.

Si se considera la siguiente solución **T**

	Ecuación 1	Ecuación 2	Ecuación 3	<i>MI</i>
$z_2$	1	3	0	1
$z_4$	1	2	0	0

ésta indica que la variable requerida  $z_2$  se calcula mediante la combinación lineal de las ecuaciones 1 y 3, en tanto que la variable requerida  $z_4$  se estima utilizando la combinación lineal de las ecuaciones 1 y 2. La última columna del arreglo indica que  $z_2$  y  $z_4$  son consideradas como variable medida no redundante y variable no medida, respectivamente.

La calidad de la población inicial puede mejorarse si la variable aleatoria discreta *MI* asume un valor -1, pues ésto permite introducir variables medidas redundantes en la primera generación. Con el objeto de incrementar la redundancia y consecuentemente la precisión, se consideran altas probabilidades para el evento  $MI=-1$ .

El tamaño máximo de la población está dado por el número total de combinaciones entre los  $(2^{J-1} nrv)$  subconjuntos de ecuaciones que pueden ser utilizadas para calcular las variables requeridas y los tres resultados posibles de la variable *MI*.

Para generar una solución inicial factible se aplica el siguiente procedimiento para cada variable requerida *irv* ( $irv = 1, \dots, nrv$ ).

- 1) Definir el conjunto  $\mathcal{E}_i = \{1, \dots, J\}$ .
- 2) Definir  $\mathcal{C}$  como el subconjunto de los elementos de  $\mathcal{E}$  que corresponden a las

filas de  $\mathbf{D}$  con entradas distintas de cero para la variable  $irv$ .

- 3) Seleccionar el primer elemento de la fila  $(T_{irv,1})$ . Se elige, con idéntica probabilidad, un elemento del conjunto  $\mathcal{C}$ .
- 4) Definir  $\mathcal{E}' = \mathcal{E} - \mathcal{C}$
- 5) Inicializar  $T_{irv,w} = 0$ , con  $w = 2, \dots, J$
- 6) Generar  $T_{irv,w}$  con  $w = 2, \dots, J$ ; se selecciona una combinación del conjunto

$\binom{|\mathcal{E}'|}{b}$ ,  $b \in \{0, 1, \dots, |\mathcal{E}'|\}$  usando un procedimiento aleatorio. Primero se

genera un valor de  $b$  y luego se obtiene un conjunto  $\mathcal{E}$  de cardinalidad  $b$ , de los elementos de  $\mathcal{E}'$ , con la siguiente metodología:

Cada ecuación se incluye en forma secuencial en la combinación propuesta, si entre las variables que participan, existe al menos una que tenga un coeficiente cuyo valor sea opuesto al de las variables de la ecuación que se está considerando. Esto impide, en una primera instancia, que una ecuación sólo introduzca variables sin quitar ninguna otra. Esto es, si  $\mathcal{E}^l$  es el conjunto de variables de la combinación lineal actual, entonces la ecuación  $j \in \mathcal{E}'$  se adiciona a la combinación lineal para dar  $\mathcal{E}^{l+1}$  si  $\mathcal{E}^{l+1} \not\subset \mathcal{E}^l$ . Esta última condición puede ser relajada al admitir con cierta probabilidad  $IC$  que la combinación lineal incorpore ecuaciones que introducen un mayor número de variables a medir. En los casos extremos, si  $IC=1$ , todas las  $b$  ecuaciones de  $\mathcal{E}$  son consideradas en la combinación lineal resultante, mientras que si  $IC=0$  lo son sólo aquellas que verifican la condición antes descrita.

- 7) Seleccionar aleatoriamente los valores de  $MI$ .

Cada solución inicial factible se codifica en un vector binario,  $\mathbf{q}$ , que representa una propuesta de instrumentación. Esto se realiza a través del siguiente procedimiento:

- 1) Inicializar el vector de variables medidas (**q**) y el vector de memoria (**mem**) como vectores nulos. Ambos vectores son de dimensión  $I$ .
- 2) Repetir para cada variable  $irv$  ( $irv=1, \dots, nrv$ ) perteneciente al vector **req** de variables requeridas
  - i) Si  $MI(irv)=1$ , la variable es medida y no redundante, el elemento del vector **q** que corresponde a  $\mathbf{req}(irv)$  es igual a 1, esto es  $\mathbf{q}(\mathbf{req}(irv))=1$
  - ii) Si  $MI(irv)=0$ , la variable es no medida, entonces su valor será calculado usando la combinación lineal propuesta en **T**. En este caso, las variables que participan en la ecuación resultante, excepto  $irv$ , deberían ser medidas e incluidas en **q**. El vector **mem** se actualiza incorporando tanto las nuevas variables medidas como las estimables.
  - iii) Si  $MI(irv)=-1$ , la variable es redundante (medida y estimable), los procedimientos i) e ii) se aplican simultáneamente.

hasta que el vector **mem** contenga unos en las posiciones correspondientes a todas las variables requeridas.

Al finalizar el procedimiento, las componentes del vector **q** serán iguales a 1 para las variables medidas, mientras las componentes del vector **mem** serán iguales a 1 si la variable correspondiente es medida o estimable.

Al conjunto de soluciones así generado se le incorpora la solución trivial del PDRS, esto es, medir todas las variables. De esta manera se garantiza la presencia de al menos

una propuesta de medición que cumpla con todas las restricciones impuestas. Las soluciones se evalúan utilizando el procedimiento que se presentará en la sección 3.3.1.3. Se considera que la mejor propuesta de solución es aquella que tiene asociado el menor valor de una cierta función. A dicha solución se la selecciona como punto inicial para el algoritmo de BT.

### 3.3.1.2. Definición de la Vecindad

Dada una solución  $\mathbf{q}$ , la vecindad de  $\mathbf{q}$  se define como el conjunto de todas las soluciones  $\mathbf{q}'$  que corresponden a agregar o quitar una medición a la propuesta corriente. Esto corresponde al conjunto de soluciones cuya distancia de Hamming respecto de  $\mathbf{q}$  es igual 1. En símbolos:

$$\mathcal{N}(\mathbf{q}) = \{ \mathbf{q}' / q'_i = \text{no } (q_i) \text{ para cualquier índice único } i \text{ y } q'_i = q_i \ \forall i \neq l \}$$

### 3.3.1.3. Función de Evaluación

Dado que un movimiento puede originar una solución no factible, una solución de la vecindad se evalúa utilizando una Función de Evaluación,  $F$ , que tiene en cuenta las violaciones a las restricciones del problema de optimización, planteado en la ecuación (3.1), de la siguiente forma:

$$F = \begin{cases} \sum_{i=1}^l c_i q_i & \text{si } \mathbf{q} \text{ es factible} \\ C(\mathbf{q}) & \text{si } \mathbf{q} \text{ es no factible} \end{cases} \quad (3.13)$$

Si se tiene en cuenta que una solución es no factible si alguna de las variables claves no medidas es no observable, o si se satisface dicha restricción pero no se alcanzan los requerimientos de precisión impuestos, la función  $C(\mathbf{q})$  propuesta contempla ambas situaciones, de la siguiente manera:

$$C(\mathbf{q}) = \begin{cases} FB + (FB - (\sum_{i=1}^I c_i q_i)) \frac{nvo}{nrv} & \text{si } \mathbf{q} \text{ no cumple restricciones de } \mathcal{S}_{\mathcal{E}} \\ FB + (\sum_{i=1}^I c_i q_i) \frac{1}{nvp} \sum_{i=1}^{nvp} \frac{\sigma_i - \sigma_i^*}{\sigma_i} & \text{si } \mathbf{q} \text{ no cumple restricciones de } \mathcal{S}_{\sigma} \end{cases} \quad (3.14)$$

donde  $\sum_{i=1}^I c_i$  es el límite superior de la función objetivo del PDRS, ya que representa el costo de la red de sensores cuando todas las variables involucradas son medidas, y se denomina Límite de Factibilidad,  $FB$ .

En la ecuación (3.14),  $nvo$  y  $nvp$  representan el número de variables en  $\mathcal{S}_{\mathcal{E}}$  y  $\mathcal{S}_{\sigma}$ , respectivamente, cuyas restricciones no se satisfacen y,  $nrv$  es el número de variables pertenecientes al conjunto  $\mathcal{S}_{\mathcal{E}}$ .

#### 3.3.1.4. Memorias de corto y largo plazo

La LTR es un vector de dimensión  $I$ . Si una de sus componentes presenta un valor distinto de cero, esto indica que el movimiento asociado está prohibido dado que fue modificado recientemente. Su valor es el número de iteraciones que restan para permitir nuevamente dicho movimiento. Esta cantidad está prefijada y se denomina período tabú ( $pt$ ).

La LTF se representa mediante un vector  $\mathbf{h}$  de dimensión  $I$ . La  $i$ -ésima componente de  $\mathbf{h}$  almacena el número de movimientos de la variable  $i$  usados para generar la próxima solución durante  $ph$  iteraciones. Si el proceso de búsqueda se estanca en un área específica, es necesario dirigir la búsqueda a regiones no visitadas o visitadas con menor frecuencia. En consecuencia el valor de la  $F$  correspondiente al  $i$ -ésimo movimiento permitido se penaliza en proporción a  $h_i$ . Luego de  $ph$  iteraciones el vector  $\mathbf{h}$  se reinicializa.

### 3.3.1.5. Criterios de Aspiración y Detención

El estado tabú asociado a un movimiento puede ser revocado si la solución generada a partir de dicho movimiento es mejor, en términos del valor de la  $F$ , que aquella correspondiente a  $\mathbf{q}^*$ .

El criterio de detención utilizado es finalizar el procedimiento cuando se han realizado un número especificado de iteraciones, *maxiter*, sin encontrar ninguna solución mejor que  $\mathbf{q}^*$ .

En la Figura 3.1 se muestra el pseudocódigo del algoritmo desarrollado para resolver el PDRS mediante la BTC.

```

Generar una solución inicial  $\mathbf{q}_0$ 
 $\mathbf{q}^* = \mathbf{q} = \mathbf{q}_0$  y  $F(\mathbf{q}^*) = F(\mathbf{q}_0)$ 
for  $l = 1$  to #maxiter do
  Genere la vecindad  $\mathcal{N}(\mathbf{q})$ 
  Seleccione  $\mathbf{q}' \in \mathcal{N}(\mathbf{q})$  con el menor valor de  $F$ 
    if  $\mathbf{q}' \in \mathcal{N}(\mathbf{q})$  satisface el criterio de aspiración  $F(\mathbf{q}') < F(\mathbf{q}^*)$ 
       $\mathbf{q}^* = \mathbf{q}'$  y  $F(\mathbf{q}^*) = F(\mathbf{q}')$ 
    else
      Seleccionar una nueva solución  $\mathbf{q}' \in \mathcal{N}(\mathbf{q})$  que minimice  $F(\mathbf{q}')$  y sea
      no tabú
    endif
  Prohibir el movimiento inverso durante  $pt$  iteraciones y actualizar  $\mathbf{h}$ 
   $\mathbf{q} = \mathbf{q}'$ 
endfor
return  $\mathbf{q}^*$ 

```

Figura 3.1. Pseudocódigo del algoritmo basado en BTC

### 3.3.2 Algoritmo basado en la Búsqueda Tabú Clásica y la Técnica de Reencadenamiento de Caminos

Tal como sucede en la mayoría de las heurísticas propuestas en la literatura, el balance apropiado entre la intensificación y diversificación de la búsqueda juega un rol central en el éxito de la metodología propuesta. En el contexto de la BTC, el procedimiento de reencadenamiento de caminos (RC) se puede incorporar con este fin. Dicha técnica fue propuesta por Glover y Laguna (2002) y ha sido reconocida como una herramienta efectiva en la resolución de problemas combinatorios (Ghamlouche y col., 2004).

Dado un conjunto de referencia  $\mathcal{R}$  construido con soluciones de alta calidad, la técnica de RC consiste básicamente en seleccionar dos soluciones de  $\mathcal{R}$ , llamadas

solución inicial ( $SI$ ) y solución guía ( $SG$ ), y generar un camino entre  $SI$  y  $SG$  seleccionando movimientos que introduzcan en la solución corriente atributos contenidos en  $SG$ . La filosofía subyacente de esta estrategia es que las soluciones de alta calidad comparten ciertos atributos y que sus combinaciones producen soluciones que podrían ser mejores que aquellas contenidas en  $\mathcal{R}$  y alguna aún mejor que la mejor encontrada hasta el momento. Su desempeño depende de las reglas adoptadas para: generar  $\mathcal{R}$ , seleccionar las soluciones  $SI$  y  $SG$ , e identificar la estructura de vecindades y los atributos guía.

Luego de un número  $freqpr$  predefinido de movimientos consecutivos, el procedimiento de BTC cambia a la fase RC, la cual se detiene cuando la cardinalidad de  $\mathcal{R}$  es menor o igual que uno ( $|\mathcal{R}| \leq 1$ ). Si la condición de detención del algoritmo no se verifica, entonces el procedimiento se repite para formar un nuevo conjunto de referencia  $\mathcal{R}$ .

El procedimiento de formación del conjunto de referencia  $\mathcal{R}$  es esencial para generar nuevas soluciones de alta calidad. En este trabajo de tesis se propone crear  $\mathcal{R}$  durante la ejecución de BTC y mejorarlo durante la fase de RC. Se utilizan las siguientes reglas para asegurar la calidad y diversidad de las soluciones contenidas en  $\mathcal{R}$  (Carnero y col., 2009).

- 1) Generar un conjunto auxiliar  $\mathcal{CA}$  a partir de las mejores soluciones encontradas en la fase BTC. Cada uno de los vectores contenidos en  $\mathcal{CA}$  se indican con  $\mathbf{q}_{CA}$ . Dada una iteración de la BTC, si es posible encontrar una solución superadora, entonces, ésta se incorpora al conjunto auxiliar  $\mathcal{CA}$ .
- 2) En la primera mitad del conjunto  $\mathcal{R}$  se almacenan los mejores vectores solución que pertenecen al  $\mathcal{CA}$ .



- 3) Para cada vector solución  $\mathbf{q}_{CA}$  que pertenece al  $\mathcal{CA}$ , pero que no está incluido en  $\mathcal{R}$ , esto es  $\mathbf{q}_{CA} \in \{\mathcal{CA} \setminus \mathcal{R}\}$ , se computa la distancia de Hamming,  $d_H$ , entre  $\mathbf{q}_{CA}$  y  $\mathcal{R}$  de la siguiente forma:

$$d_H(\mathbf{q}_{CA}, \mathcal{R}) = \text{Min} \sum_i^l |q_i^l - q_{CAi}^l| \quad \forall l \in \mathcal{R} \quad (3.15)$$

- 4) La segunda mitad de  $\mathcal{R}$  se completa con el vector  $\mathbf{q}_{CA} \in \{\mathcal{CA} \setminus \mathcal{R}\}$  que maximiza  $d_H$ .

En relación con  $SI$  y  $SG$ , éstas se definen como la peor y la mejor solución de  $\mathcal{R}$ , respectivamente. Para realizar el procedimiento de RC se han propuesto dos tipos de movimientos denominados M1 y M2, que difieren en los atributos que guían la trayectoria y, en consecuencia, en la estructura de la vecindad que generan.

Estos movimientos se presentan a continuación y en la Figura 3.2 se muestra un pseudocódigo para el algoritmo de RC implementado.

### 3.3.2.1. Movimiento M1

Supóngase que  $\mathbf{q}_i$  y  $\mathbf{q}_g$  representan las soluciones  $SI$  y  $SG$  actuales, respectivamente. Partiendo desde  $\mathbf{q}_i$ , la idea es generar vecinos que tiendan a  $\mathbf{q}_g$ . Para formar el primer vecino  $\mathbf{q}_1$ , inicialmente éste se define igual a  $\mathbf{q}_i$ , luego se comparan los elementos de los vectores  $\mathbf{q}_i$  y  $\mathbf{q}_g$ , de izquierda a derecha, hasta que se encuentra la primera diferencia. Para esta posición, por ejemplo la  $l$ -ésima componente de los vectores, el elemento de  $\mathbf{q}_i$  se reemplaza por el correspondiente elemento de  $\mathbf{q}_g$  y la comparación se detiene dado que se ha formado el primer vecino. A continuación se repite el mismo procedimiento entre  $\mathbf{q}_1$  y  $\mathbf{q}_g$  para obtener  $\mathbf{q}_2$ , pero la comparación comienza en la posición  $(l+1)$  en este caso.

El resto de los vecinos se generan del mismo modo, y el procedimiento finaliza cuando el último vecino formado difiere sólo en una componente respecto de  $\mathbf{q}_g$ .

### 3.3.2.2. Movimiento M2

La estructura de vecindades propuesta se basa en el intercambio de posición de una medición. Se ha observado que las colecciones de buenas soluciones presentan a menudo el mismo número de mediciones aunque la distribución de las mismas difiere en alguna posición del diagrama de flujo del proceso. Por tal motivo se propone que partiendo desde la solución  $SI$  se generen sucesivas vecindades que contengan soluciones, con igual número de mediciones, cada vez más similares a la solución  $SG$ .

En primer lugar, se calcula la diferencia entre los vectores  $\mathbf{q}_g$  y  $\mathbf{q}_i$ , obteniéndose valores positivos (+1), negativos (-1) o cero para las componentes del vector diferencia. Las posiciones con valores positivos y negativos se registran en los vectores  $\mathbf{pp}$  y  $\mathbf{np}$ , respectivamente. Luego, todas las combinaciones entre cada elemento de  $\mathbf{pp}$  y todos los elementos de  $\mathbf{np}$  conforman un conjunto de intercambios entre los correspondientes elementos de  $\mathbf{q}_i$ . Cada intercambio genera un vecino.

Por ejemplo si,  $SG = [1 \ 1 \ 1 \ 0 \ 0]$  y  $SI = [1 \ 0 \ 0 \ 1 \ 1]$ , entonces  $\mathbf{pp} = [2 \ 3]$  y  $\mathbf{np} = [4 \ 5]$ . Los posibles vecinos para la solución inicial utilizando M2 serán:  $[1 \ 1 \ 0 \ 0 \ 1]$ ,  $[1 \ 1 \ 0 \ 1 \ 0]$ ,  $[1 \ 0 \ 1 \ 0 \ 1]$ ,  $[1 \ 0 \ 1 \ 1 \ 0]$ . Cada una de estas soluciones tiene una estructura más similar a la solución guía, hasta alcanzarse  $SG$  en la fase final. Si las soluciones, inicial y guía, no comparten el mismo número de mediciones, el tipo de movimiento puede utilizarse aún para proporcionar soluciones muy cercanas a  $SG$  en el final de la trayectoria de RC.

```

 $\mathcal{R}$ , conjunto de referencia;  $\mathbf{q}^*$ , mejor solución actual
Seleccionar la solución inicial  $\mathbf{q}_i$  y la solución guía,  $\mathbf{q}_g$ 
 $\mathbf{q} = \mathbf{q}_i$ 
while  $\mathbf{q} \neq \mathbf{q}_g$ 
    Generar vecindad  $\mathcal{N}_1(\mathbf{q})$  (M1)
    Generar vecindad  $\mathcal{N}_2(\mathbf{q})$  (M2)
    Seleccionar una solución  $\bar{\mathbf{q}} \in \mathcal{N}_1(\mathbf{q}) \cup \mathcal{N}_2(\mathbf{q})$  tal que minimice  $F(\bar{\mathbf{q}})$  o satisfaga
    el criterio de aspiración
     $\mathbf{q} = \bar{\mathbf{q}}$ 
return  $\mathbf{q}$ 

```

Figura 3.2. Pseudocódigo para el algoritmo de RC

### 3.4 Nuevas Estrategias de Diseño de Redes de Sensores basadas en Búsqueda Dispersa

La Búsqueda Dispersa es un enfoque evolutivo basado en el principio de que los métodos sistemáticos usados para crear nuevas soluciones brindan beneficios superiores a los producidos cuando las soluciones se forman de manera aleatoria.

La descripción algorítmica de esta metaheurística fue presentada por Glover en 1998. En esta versión se genera un conjunto inicial de soluciones que garantiza cierto nivel de diversidad, y luego se aplican procedimientos heurísticos, diseñados para el problema bajo consideración, para intentar mejorar este conjunto de soluciones.

Inicialmente se selecciona un conjunto de referencia,  $\mathcal{R}_{ref}$ , constituido por un subconjunto de las mejores soluciones obtenidas en términos de calidad y diversidad. Luego se generan nuevas soluciones, aplicando combinaciones estructuradas entre subconjuntos de  $\mathcal{R}_{ref}$ , y se aplican procedimientos de mejora con el fin de optimizarlas. Posteriormente se agregan a  $\mathcal{R}_{ref}$  una colección de las mejores soluciones encontradas

en el procedimiento de mejora. Estas etapas se repiten hasta que  $P_{\text{objet}}$  permanece sin cambios.

El algoritmo propuesto por Glover se compone de cuatro partes destinadas a ejecutar las siguientes tareas: Diversificación-Generación, Mejora de Soluciones, Generación y Actualización de  $P_{\text{objet}}$ , Generación de Subconjuntos y Combinación de Soluciones. En la siguiente subsección se describen los métodos desarrollados para llevar cabo estas tareas con el fin de resolver el PDRS (Carnero y col., 2006, 2009).

### 3.4.1 Algoritmo basado en Búsqueda Dispersa

#### 3.4.1.1. Generación-Diversificación

Para generar una colección de soluciones de prueba se utilizó el mismo método descrito en 3.3.1.1. Cada solución garantiza que todas las variables claves son estimables. Sin embargo, en este conjunto existen tanto soluciones factibles como no factibles, esto garantiza un cierto nivel de diversidad, que es necesario para mejorar la capacidad exploratoria de la técnica.

Las soluciones más diversas con respecto a la  $F$  definida por la ecuación 3.13 se incorporan en un conjunto  $\mathcal{U}$  de tamaño  $U_s$ , donde  $U_s$  es un parámetro del algoritmo. Las soluciones se ordenan de acuerdo al valor de la  $F$ . El conjunto  $\mathcal{U}$  se forma con la colección de las  $U_s/2$  soluciones con menor valor de la  $F$  y las  $U_s/2$  peores soluciones.

#### 3.4.1.2. Mejora de Soluciones

Se diseñaron dos procedimientos para mejorar cada una de las soluciones incluidas en  $\mathcal{U}$ .

- *Método de Mejora 1 (MM1)*

El *MM1* consiste en una búsqueda local que inspecciona una vecindad definida por el conjunto de soluciones que se alcanzan desde la actual mediante el intercambio de dos variables. Si  $\mathcal{S}$  representa el conjunto de sensores de una solución, entonces cada elemento de  $\mathcal{S}$  puede reemplazarse por un instrumento perteneciente al conjunto  $\mathcal{S}/\mathcal{S}$ , siendo  $\mathcal{S}$  el conjunto de todos los sensores disponibles. El tamaño de la vecindad y por lo tanto el esfuerzo computacional asociado a este mecanismo de mejora es elevado para problemas de dimensiones considerables. Por esta razón no se aplica a todas las soluciones sino que sólo se utiliza en dos situaciones: como rutina de mejora para las soluciones de la fase de generación de soluciones diversas y cuando el mecanismo alternativo de mejora no logra incorporar nuevas soluciones a *Refinet*.

- *Método de Mejora 2 (MM2)*

El *MM2* se desarrolla con el fin de identificar bloques constructivos que comparten las soluciones de alta calidad, y generar vecindades de las mismas que respeten dichas estructuras. Para ello se comparan pares de soluciones, y se halla un esquema de solución parcial a partir del cual se construye la vecindad. Ésta se forma con el conjunto de todas las soluciones completas alcanzables desde la parcial.

Por ejemplo si  $\mathbf{q}_1 = [1\ 1\ 1\ 1\ 1\ 0]$  y  $\mathbf{q}_2 = [0\ 1\ 1\ 1\ 1\ 0]$  son soluciones obtenidas por combinación, entonces el esquema parcial es  $[* 1\ 1\ 1\ 1\ **]$ , y la vecindad generada a partir de él es el conjunto:

$$\mathcal{N}(\mathbf{q}_1, \mathbf{q}_2) = \{[0\ 1\ 1\ 1\ 1\ 0\ 0]; [0\ 1\ 1\ 1\ 1\ 1\ 0]; [0\ 1\ 1\ 1\ 1\ 1\ 1]; [1\ 1\ 1\ 1\ 1\ 0\ 0];$$

$$[1\ 1\ 1\ 1\ 1\ 0\ 1]; [1\ 1\ 1\ 1\ 1\ 1\ 1]\}.$$

### 3.1.1.1 Generación y Actualización del Conjunto de Referencia

La construcción del conjunto de referencia se realiza mediante la selección de  $Rf$  soluciones contenidas en  $U$ . La mitad de ellas corresponden a las  $Rf/2$  soluciones de mejor valor de  $F$ . Los restantes elementos del conjunto se eligen entre las soluciones más diversas contenidas en  $U$  teniendo en cuenta su  $d_H$  tal como se detalló en la subsección 3.3.2.

### 3.1.1.2 Generación de Subconjuntos y Combinación de Soluciones

La combinación de soluciones es una operación que se define sobre un subconjunto de  $\mathcal{P}_{\text{Ref}}$ . El criterio de formación de subconjuntos utilizado fue el de parejas teniendo en cuenta que los métodos de combinación propuestos resultan en operadores que requieren dos soluciones como argumento de entrada. Para cada par de soluciones de  $\mathcal{P}_{\text{Ref}}$  se implementaron dos mecanismos de combinación diferentes, sugeridos en la literatura (Glover y Laguna, 2002), con el objetivo de comparar su desempeño. A continuación se detallan los aspectos fundamentales de los mismos.

#### *Mecanismo de combinación 1 (MCI)*

- 1) considerar a cada variable binaria de la solución combinada,  $q_i$ , como una variable aleatoria discreta  $X_i$ .
- 2) asignar a la variable aleatoria  $X_i$  la siguiente distribución de probabilidades:

$$\Pr(X_i = 1) = \frac{F(l_1) \cdot q_i^{l_1} + F(l_2) \cdot q_i^{l_2}}{F(l_1) + F(l_2)} \quad (3.16)$$

$$\Pr(X_i = 0) = 1 - \Pr(X_i = 1)$$

En la expresión anterior,  $q_i^{l_1}$  y  $q_i^{l_2}$  representan el valor de la  $i$ -ésima variable binaria en las soluciones  $l_1$  y  $l_2$  respectivamente. Éstas forman el par de soluciones pertenecientes a  $\mathcal{P}_{\text{Ref}}$  que fueron seleccionadas para combinación.

- 5) generar una muestra aleatoria de tamaño uno de la variable  $X_i$ . La probabilidad asignada a  $X_i$  se compara con un número aleatorio  $nal$ , y se define el valor de  $q_i$  en la solución combinada utilizando el siguiente criterio:

$$q_i = \begin{cases} 1 & \text{si } n_{al} \leq \Pr(X_i = 1) \\ 0 & \text{si } n_{al} > \Pr(X_i = 1) \end{cases} \quad (3.17)$$

De las ecuaciones anteriores se deduce que: 1) si las componentes homólogas de las soluciones de partida son iguales, el valor de  $q_i$  de la solución combinada es igual al de aquellas; 2) si las componentes homólogas de las soluciones de partida son diferentes, una buena solución tiene mayor probabilidad de copiar sus componentes en la solución combinada dado que el numerador de la ecuación 3.16 se pondera utilizando el valor binario de la componente correspondiente.

#### *Mecanismo de combinación 2 (MC2)*

El *MC2* corresponde a un cruzamiento uniforme. Por lo tanto no introduce un sesgo hacia la solución de referencia de mejor calidad tal como sucede al aplicar el *MCI*.

Este mecanismo consiste en intercambiar todas las componentes homólogas entre las dos soluciones seleccionadas para combinación con una probabilidad del 50%. Se genera una máscara en forma aleatoria de la misma dimensión que las soluciones involucradas y se procede de la siguiente manera: si el bit  $i$  de la máscara es 1, se copia la componente  $i$  de la primera solución considerada, de lo contrario la solución combinada hereda en la posición  $i$  el valor correspondiente a la segunda solución.

En la Figura 3.3 se describe el pseudocódigo del algoritmo de BD propuesto para resolver el PDRS.

```

U =  $\emptyset$ 
U = método de Generación- Diversificación (Us)
U = método de Mejora I (U)
RefSel = método de Generación y Actualización de RefSel (U, Rf)
RefSel = Sort(RefSel)
 $\mathbf{q}_{\text{best}}$  = solución de menor valor de F
 $\mathbf{q}_{\text{worst}}$  = solución de mayor valor de F
Nueva Solución = TRUE
while (Nueva Solución)
    Nueva Solución = FALSE
    Generar subconjuntos de RefSel, usando el método de Generación de Subconjuntos
    while (exista al menos un subconjunto sin evaluar)
        Seleccionar el subconjunto y rotular como evaluado
        Aplicar métodos de combinación de soluciones en el subconjunto
        Aplicar método de Mejora 2 a cada solución obtenida por combinación. Sea
         $\mathbf{q}$  la solución mejorada,
        if F( $\mathbf{q}$ ) es mejor que F( $\mathbf{q}_{\text{worst}}$ ) y  $\mathbf{q}$  no está incluida en RefSel
             $\mathbf{q}_{\text{worst}} = \mathbf{q}$ 
            RefSel = Sort(RefSel)
            Nueva Solución = TRUE
        else
             $\mathbf{q}$  = método de Mejora I( $\mathbf{q}$ )
            if F( $\mathbf{q}$ ) <  $\mathbf{q}_{\text{best}}$  solución en RefSel
                 $\mathbf{q}_{\text{best}} = \mathbf{q}$ 
                NuevaSolución = TRUE
            endif
        endif
    endwhile
endwhile

```

Figura 3.3. Pseudocódigo para el algoritmo de Búsqueda Dispersa

### 3.5 Resultados de Aplicación

Los algoritmos propuestos fueron aplicados al diseño de la red de caudalímetros másicos del sistema de vapor de una planta de producción de metanol presentado por Serth y Heenan (1986). En esta tesis dicho proceso se representa mediante las siglas SMN. El mismo consta de 28 corrientes y 11 equipos. Los casos de estudio consideran que los



sensores pueden ubicarse sobre cualquier corriente, suposición que aumenta el número de combinaciones a evaluar por el algoritmo de resolución. El espacio de búsqueda para este problema es  $2^{28}=268.435.456$  combinaciones posibles.

### 3.5.1 Descripción del Proceso

El diagrama de flujo del SMN se esquematiza en la Figura 3.4. Además, los valores verdaderos de los caudales máxicos,  $C^m_i$  ( $i=1\dots I$ ), expresados en t/h se reportan en la Tabla 3.1. La misma incluye el desvío estándar del error de la medición del  $i$ -ésimo caudalímetro máxico,  $\sigma_i$ , y su costo,  $c_i$ . Esta información se extrajo del trabajo de Sen y col. (1998).

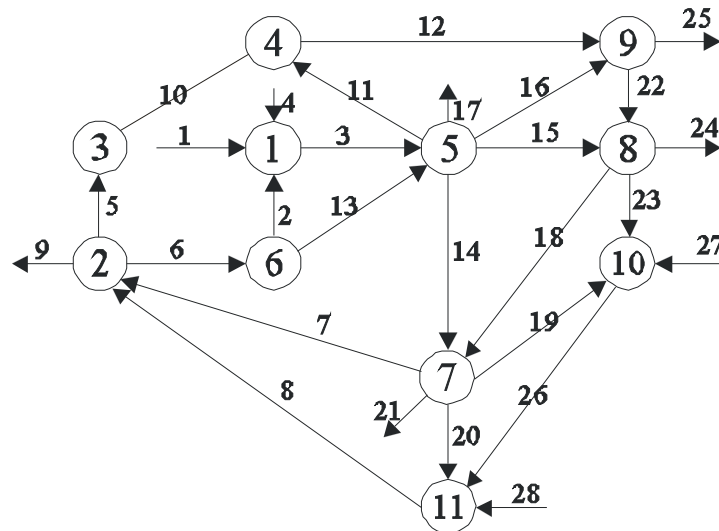


Figura 3.4. Diagrama de Flujo del SMN

Corriente	$C^m_i$	$\sigma_i$	$c_i$
1	0.86	0.0215	3.7
2	1.	0.025	4.5
3	111.82	2.8	132.2
4	109.95	2.749	129.2
5	53.27	1.332	65.3
6	112.27	2.807	132.4
7	2.32	0.058	5.0
8	164.05	4.101	193.9
9	0.86	0.0215	2.06
10	52.41	1.31	62.8
11	14.86	0.3715	20.2
12	67.27	1.682	80.0
13	111.27	2.782	130.4
14	91.86	2.296	109.8
15	60.	1.5	71.6
16	23.64	0.591	29.7
17	32.73	0.8182	39.5
18	16.23	0.4057	20.4
19	7.95	0.1987	11.1
20	10.5	0.2625	13.6
21	87.27	2.182	102.9
22	5.45	0.1362	8.1
23	2.59	0.0648	6.3
24	46.64	1.166	55.5
25	85.45	2.136	101.0
26	81.32	2.033	93.7
27	70.77	1.769	84.7
28	72.23	1.806	85.4

Tabla 3.1. Datos del SMN

### 3.5.2 Casos de Estudio

Los algoritmos presentados en este capítulo se utilizaron para resolver dos PDRS de distinta complejidad, denominados Diseño 1 y Diseño 2, para el proceso descrito previamente.

Para el Diseño 1 el conjunto de variables requeridas sujetas a restricciones de estimabilidad es  $\mathcal{S}_\sigma = \{z_1, z_2, z_6\}$ , mientras que el subconjunto  $\mathcal{S}_\sigma = \{z_2, z_6\}$  corresponde a las variables sujetas también a restricciones de precisión. Dado que las variables claves pertenecen a una misma región en el diagrama de flujo, si bien el proceso consta de 28 corrientes, varias de ellas no intervienen en la estimabilidad de las variables requeridas y de algún modo se espera que esto disminuya la complejidad del diseño a resolver. En el Diseño 2, se mantiene el número de variables sujetas a restricciones de estimabilidad y precisión, pero se consideraron corrientes muy alejadas físicamente entre sí. En este caso, el número de potenciales soluciones a explorar aumenta ya que se incrementa el número de balances de masa que pueden plantearse para satisfacer la observabilidad de las diferentes variables claves. Para el Diseño 2,  $\mathcal{S}_\sigma = \{z_2, z_{10}, z_{28}\}$  y  $\mathcal{S}_\sigma = \{z_{10}, z_{28}\}$ .

La selección de los parámetros utilizados para la ejecución de los algoritmos se ha basado en diferentes criterios. Así, para el caso de los valores asignados a la distribución de  $MI$ , éstos permiten obtener soluciones iniciales con algún sesgo hacia la redundancia sobre las variables claves, con el objeto de contar, en principio, con estimaciones que tengan una precisión no tan alejada de la cota impuesta. Por otra parte, para los períodos  $pt$  y  $ph$  los valores concuerdan con aquellos sugeridos habitualmente en la literatura. El número máximo de iteraciones ha sido ajustado por experimentación, así como el tamaño de los conjuntos de referencia para los algoritmos RC y BD.

A continuación se detallan los valores asignados a los parámetros y se muestran y discuten los resultados obtenidos al aplicar los algoritmos basados en BTC, BTC con RC (BTC-RC) y BD para el Diseño 1 y Diseño 2.

### 3.5.3 Resultados

#### 3.5.3.1. Algoritmo basado en BTC

En la Tabla 3.2 se listan los valores seleccionados para los parámetros del algoritmo.

Parámetro	Valor
Probabilidad $MI = -1$	0.4
Probabilidad $MI=1$	0.3
$IC$	0
$pt$	5
$ph$	25
$maxiter$	200

Tabla 3.2. Parámetros del algoritmo basado en BTC

En la Tabla 3.3 se muestran los datos y resultados para el Diseño 1. Para cada elemento  $le$  del conjunto  $\mathcal{S}_e$ , indicado mediante el índice del correspondiente flujo másico, se presentan: el valor del límite superior de la restricción de precisión  $\sigma^*$ , y el valor de la misma restricción en la solución,  $\hat{\sigma}$ . Además se reporta el vector solución  $\mathbf{q}_{sol}$ , también expresado mediante el índice de los caudales másicos medidos, y el valor de la  $F$  para  $\mathbf{q}_{sol}$ .

Variables claves	$\sigma_i^*$	$\hat{\sigma}_i$	$\mathbf{q}_{sol}$	$F$
1	-----	$\hat{\sigma}_1=0.0215$	1 2 6 7 9 10 13 20 26 28	533.56
2	$\sigma_2^*=0.025$	$\hat{\sigma}_2=0.025$		
6	$\sigma_6^*=1.7851$	$\hat{\sigma}_6=1.6552$		

Tabla 3.3. Datos y Resultados para el Diseño 1 - BTC

Se analizó además la sensibilidad del algoritmo con respecto a la solución inicial, teniendo en cuenta que ésta puede variar dada la aleatoriedad que se introduce a través de la variable  $MI$  en la generación del conjunto inicial. Con este objetivo se realizaron 100 corridas diferentes del procedimiento propuesto y se evaluaron las soluciones obtenidas. Se calcularon el valor mínimo, la media, el desvío estándar y el porcentaje del coeficiente de variación (CV%) de las soluciones. Este último representa la variación porcentual relativa a la media obtenida y permitirá comparar la heterogeneidad de los valores alcanzados para los diferentes casos de diseño. Los estadísticos antes mencionados se incluyen en la Tabla 3.4, donde se observa que el algoritmo BTC es capaz de alcanzar, en promedio, soluciones con un valor cercano al mínimo de referencia de 533.56.

Mínimo	Media	Desvío Estándar	CV[%]
533.56	549.39	39.49	7.19

Tabla 3.4. Estadísticos de la solución para el Diseño 1 - BTC

En la Tabla 3.5 se reportan los datos y resultados para el Diseño 2. Además los estadísticos de las soluciones obtenidas al realizar 100 experiencias con diferentes soluciones iniciales se proporcionan en la Tabla 3.6. Puede observarse que el desempeño del algoritmo depende fuertemente del punto inicial, dado que sólo encuentra el mejor valor en 24 experiencias sobre un total de 100 corridas. La solución inicial factible más simple es aquella en la cual se colocan sensores en todas las corrientes, sin embargo cabe

mencionar que ésta no resulta un buen punto inicial, para este caso de diseño particular, dado que el algoritmo BTC no puede mejorarla.

VARIABLES CLAVES	$\sigma_i^*$	$\hat{\sigma}_i$	$\mathbf{q}_{sol}$	$F$
2	-----	$\hat{\sigma}_2=0.025$	1 2 5-10 13 19 20 23 26-28	894.86
10	$\sigma_{10}^*=1.0482$	$\hat{\sigma}_{10}=0.8861$		
28	$\sigma_{28}^*=1.4446$	$\hat{\sigma}_{28}=1.435$		

Tabla 3.5. Datos y Resultados para el Diseño 2 – BTC

Mínimo	Media	Desvío Estándar	CV[%]
894.86	1575.75	387.46	24.59

Tabla 3.6. Estadísticos de la solución para el Diseño 2 - BTC

**3.5.3.2. Algoritmo basado en BTC-RC**

En este apartado se repite el análisis presentado en la sección anterior para los Diseños 1 y 2 utilizando como procedimiento de solución el algoritmo basado en BTC-RC. Los parámetros se fijaron en los valores indicados en la Tabla 3.7

Parámetro	Valor
Probabilidad $MI = -1$	0.4
Probabilidad $MI=1$	0.3
$pt$	5
$ph$	25
$ S $	10
$frecpr$	25
$maxiter$	200

Tabla 3.7. Parámetros del algoritmo basado en BTC-RC

Los resultados presentados en las Tablas 3.3 y 3.5 son coincidentes con los obtenidos al emplear el algoritmo BTC-RC, es decir, la  $F$  se mantiene en 533.56 para el Diseño 1 y en 894.86 para el Diseño 2.

Además se ejecutaron 100 experiencias con el objetivo de analizar el comportamiento del algoritmo frente a variaciones en el conjunto inicial de soluciones. Los estadísticos de las soluciones obtenidas se reportan en las Tablas 3.8 y 3.9 para el Diseño 1 y 2 respectivamente. En el primer caso los resultados son levemente superiores a aquellos obtenidos con la BTC. Para el Diseño 2, la optimización sigue presentando dificultades respecto del punto inicial, sin embargo el empleo de la técnica RC permite hallar la mejor solución partiendo de aquella en la que todas las variables se consideran medidas, a diferencia del algoritmo basado en BTC.

Mínimo	Media	Desvío Estándar	CV[%]
533.56	543.53	32.85	6.04

Tabla 3.8. Estadísticos de la solución para el Diseño 1 – BTC-RC

Mínimo	Media	Desvío Estándar	CV[%]
894.86	1575.3	386.67	24.55

Tabla 3.9. Estadísticos de la solución para el Diseño 2 – BTC-RC

### 3.5.3.3. Algoritmo basado en BD

Se realizaron experiencias similares a las anteriores utilizando el algoritmo basado en BD como estrategia de optimización. Los parámetros del algoritmo se detallan en la Tabla 3.10. Los valores de los estadísticos calculados para las soluciones correspondientes a 100 experiencias son comparables a los obtenidos previamente para el Diseño 1, sin embargo se observa una mejora de los mismos para el Diseño 2.

Parámetro	Valor
Probabilidad $MI = -1$	0.4
Probabilidad $MI=1$	0.3
$Us$	50
$ P_{ind} $	10

Tabla 3.10. Parámetros del algoritmo basado en BD

Mínimo	Media	Desvío Estándar	CV[%]
533.56	557.9	51.9	9.3

Tabla 3.11. Estadísticos de la solución para el Diseño 1 – BD

Mínimo	Media	Desvío Estándar	CV[%]
894.86	1006.3	180.13	17.9

Tabla 3.12. Estadísticos de la solución para el Diseño 2 – BD

### 3.6 Conclusiones

En este capítulo se analizó la formulación del PDRS que satisface restricciones de estimabilidad y precisión sobre un conjunto de variables claves. Además se presentaron estrategias de resolución basadas en búsqueda local y una metodología híbrida.

El enfoque propuesto para el modelado matemático del problema no admite expresar las restricciones de estimabilidad y precisión como una función explícita de las variables binarias, ya que las dimensiones de las matrices involucradas en el modelo del proceso dependen del conjunto de mediciones propuesto, es decir del vector binario  $\mathbf{q}$ . De acuerdo a la clasificación propuesta por Nemhauser (Nemhauser y Wolsey, 1999), esto implica que el problema así modelado sea clasificado como uno de optimización combinatoria genérico y no como uno de programación no lineal mezcla entera.



Las metodologías propuestas en este capítulo son metaheurísticas basadas en BT y BD. En primer lugar se presentó un algoritmo de BTC al que luego se le incorporó la técnica de RC con el fin de darle mayor capacidad de diversificación al método. Para la BD se analizaron mecanismos de búsqueda local diferentes a los empleados en la BTC, y se incorporan a la heurística de búsqueda operaciones de combinaciones de soluciones.

Las metodologías de resolución desarrolladas se aplicaron a dos tipos de diseño óptimo diferentes, ambos correspondientes a una planta con 28 corrientes y 11 equipos. Se observó que, aún para una misma planta, las especificaciones de diseño inciden en el resultado de la optimización. Por otra parte, es de esperar que si el número de variables binarias se incrementa (plantas de mayor tamaño), la resolución se torne más compleja.

Los algoritmos basados en BTC y BTC-RC resuelven con facilidad el Diseño 1, sin embargo carecen de la robustez necesaria para abordar el Diseño 2. Esto se evidencia en la dificultad que presentan para reproducir el mejor valor encontrado, cuando el punto de inicio se modifica. En cambio la BD mejora este inconveniente. Hay que resaltar el hecho de que esta técnica híbrida, si bien utiliza mecanismos de búsqueda local, maneja un conjunto de potenciales soluciones, junto con reglas para combinar las mismas. Es decir, presenta características similares a los denominados algoritmos poblacionales que se tratan en el capítulo siguiente.

### 3.7 Nomenclatura

**A** Jacobiano de las ecuaciones del modelo respecto de las variables medidas

**B** Jacobiano de las ecuaciones del modelo respecto de las variables no medidas

---

<b>C</b>	Función de penalización
<b>c</b>	Vector de costos de instrumentación.
<b>D</b>	Matriz de dimensión $(J \times I)$ que comprende a las submatrices <b>A</b> y <b>B</b>
<b>d</b>	Vector de constantes de dimensión $J$
$d_H$	Distancia de Hamming
$E_{le}$	Grado de estimabilidad de la variable $le$
<b>F</b>	Función de evaluación
<b>FB</b>	Límite de factibilidad
$frecpr$	Frecuencia de aplicación de RC
<b>h</b>	Lista tabú basada en la frecuencia
<b>I</b>	Número de variables de proceso
<b>IC</b>	Probabilidad de incluir una ecuación en <b>T</b>
<b>J</b>	Número de ecuaciones del modelo del proceso
$maxiter$	Número máximo de iteraciones
<b>mem</b>	Vector de memoria
<b>MI</b>	Índice de Medición
$\mathcal{N}$	Vecindad de las posibles soluciones
$nvo$	Número de variables en $\mathcal{S}_e$ cuyas restricciones no se satisfacen

---

$nvp$	Número de variables en $\mathcal{L}_\sigma$ cuyas restricciones no se satisfacen
$nrv$	Número de variables requeridas
$pt$	Período tabú para la memoria de corto plazo
$\mathbf{q}$	Vector de variables binarias
$r$	Rango matricial
$\mathbf{req}$	Vector de variables requeridas
$Rf$	Cardinalidad del conjunto de referencia para BD
$SG$	Solución guía para RC
$SI$	Solución inicial para RC
$\mathbf{S}_u$	Matriz de covarianza de las estimaciones de las variables no medidas
$\mathbf{S}_x$	Matriz de covarianza de las estimaciones de las variables medidas
$\mathbf{S}_y$	Matriz de covarianza de los errores aleatorios de las mediciones
$\mathcal{L}_\sigma$	Conjunto de variables de proceso sujetas a restricciones de precisión
$\mathcal{L}_\xi$	Conjunto de variables de proceso sujetas a restricciones de estimabilidad
$\mathbf{t}$	Lista tabú basada en lo reciente
$\mathbf{T}$	Matriz de representación de soluciones
$\mathbf{u}$	Vector de variables no medidas

---

$U_s$	Cardinalidad del conjunto inicial para BD
$\mathbf{x}$	Vector de variables medidas
$X$	Variable aleatoria
$\mathbf{z}$	Vector de caudales máxicos

#### Letras Griegas

$\boldsymbol{\varepsilon}$	Vector de errores aleatorios de las mediciones
$\sigma_i$	Desvío estándar de la $i$ -ésima medición
$\hat{\sigma}_{lp}$	Desvío estándar de la estimación de la $lp$ -ésima variable
$\sigma_{lp}^*$	Desvío estándar superior de la estimación de la $lp$ -ésima variable

### 3.8 Acrónimos

BD	Búsqueda Dispersa
BT	Búsqueda Tabú
BTC	Búsqueda Tabú Clásica
LTR	Lista Tabú basada en lo Reciente
LTF	Lista Tabú basada en la Frecuencia
M1	Movimiento 1

M2	Movimiento 2
MC1	Mecanismo de Combinación 1
MC2	Mecanismo de Combinación 2
MM1	Método de Mejora 1
MM2	Método de Mejora 2
PDRS	Problema de Diseño de Redes de Sensores
RC	Procedimiento de Reencadenamiento de Caminos

### 3.9 Referencias

- Bagajewicz, M., Sánchez, M. Cost Optimal Design and Upgrade of Non-Redundant and Redundant Linear Sensor Networks. *AIChE J.*, **45**, 1927-1938, (1999).
- Carnero M, Hernández J, Sánchez M. Design of sensor networks for chemical plants based on metaheuristics. *Algorithms* **2**, 259-281, (2009).
- Carnero, M., Hernández, J., Sánchez, M. Búsqueda Dispersa Aplicada al Diseño de Redes de Sensores. *XXII Congreso Interamericano de Ingeniería Química/V Congreso Argentino de Ingeniería Química*. Disponible en CD, (2006).
- Carnero, M., Hernández, J., Sánchez, M. Optimal Sensor Network Design and Upgrade using Tabu Search. *European Symposium on Computer Aided Process Engineering – 15. L Puigjaner and A. Espuña (Editors)*. Elsevier, 1447-1452, (2005a).
- Carnero, M., Hernández, J., Sánchez, M. A Tabu Search procedure for sensor structure optimisation. *Computer Aided Optimum Design in Engineering IX.* WIT Press, 35-44. Southampton, Boston, (2005b).
- Carnero, M.; Hernández, J.; Sánchez, M.; Bandoni J. On the Solution of the Instrumentation Selection Problem. *Ind. Eng. Chem. Res.*, **44**, 2, 358-367, (2005c)

- Cavin, L., Fischer, U., Glover, F.; Hungerbühler, K. Multi-objective Process Design in Multi-purpose Batch Plants using a Tabu Search Optimization Algorithm. *Comp. Chem. Eng.*, **28**, 459-478, (2004).
- Ghamlouche, I.; Crainic, T.G.; Gendreau, M. Path relinking, cycle-based neighbourhoods and capacitated multicommodity network design. *Annals of Operations Research*, **131**, 109-133, (2004).
- Glover, F., A Template for Scatter Search and Path Relinking. *Artificial Evolution, Lecture Notes in Computer Science* 1363. J.-K. Hao, E. Lutton, E. Ronald, M. Schoenauer and D. Snyers (Eds.), Springer-Verlag, 13-54, (1998).
- Glover, F.; Laguna M. *Tabu Search*. 6th Ed. Kluwer Academic Publishers, (2002).
- Laguna, M.; Hossell, K.P.; Martí, R. *Scatter Search: Methodology and Implementation in C*; Kluwer Academic Publishers: Norwell, (2003).
- Lin, B.; Miller, D.C. Solving Heat Exchanger Network Synthesis Problems with Tabu Search. *Comp. Chem. Eng.*, **28**, 1451-1464, (2004a).
- Lin, B.; Miller, D.C. Tabu Search Algorithm for Chemical Process Optimization. *Comp. Chem. Eng.*, **28**, 2287-2306, (2004b).
- Martí, R., Laguna, M., Glover, F. Principles of Scatter Search. *European Journal of Operational Research*, **169**, 359-372, (2006).
- Nemhauser, G., Wolsey, L. *Integer and Combinatorial Optimization*, Wiley Interscience, New York, (1999).
- Sánchez, M. Romagnoli, J. Use of Orthogonal Transformations in Classification/Data Reconciliation, *Computers and Chemical Engineering*, **20**, 483-493, (1996).
- Sen S, Narasimhan S, Deb K. Sensor network design of linear processes using genetics algorithms. *Comput. Chem. Eng.*, **22**, 385-90, (1998).
- Serth, R. Heenan, W. Gross Error Detection and Data Reconciliation in Steam-Metering Systems. *AIChE J.*, **32**, 733-741, (1986).

# ***Capítulo 4***

## **Algoritmos de Estimación de Distribuciones**

## 4.1 Introducción

En este capítulo se realiza una reseña y se describen los aspectos fundamentales de los Algoritmos de Estimación de Distribuciones, AEDs, con miras a su aplicabilidad en la solución del PDRS que nos ocupa en esta tesis. Esta metodología de resolución, si bien pertenece al conjunto de técnicas denominadas Algoritmos Evolutivos (AEs), tiene como característica distintiva que la evolución se realiza mediante la construcción, actualización y posterior simulación del modelo probabilístico que mejor represente la población de potenciales soluciones del problema de optimización a resolver.

En la literatura se encuentra una amplia variedad de AEDs que presentan diferentes enfoques acerca de la construcción del modelo de distribución de probabilidades, desde modelos simples que presuponen independencia entre las variables intervinientes, hasta modelos complejos que tratan de captar y modelar las interrelaciones existentes entre las variables. De hecho, esta etapa constituye el verdadero cuello de botella de esta metodología.

A continuación se explican las características fundamentales de los AEDs y sus diferentes modelos probabilísticos, se los emplea para resolver los PDRS propuestos en el capítulo anterior y se analizan los resultados alcanzados.

Al final de este capítulo se introduce un apéndice donde se esbozan los conceptos fundamentales asociados a la teoría de modelos gráficos probabilísticos que son utilizados en la construcción de los AEDs.



## 4.2 Algoritmos de Estimación de Distribuciones

Los AEs son procedimientos estocásticos que mantienen una población de individuos  $\mathcal{S}^t = \{\mathbf{q}_1^t, \dots, \mathbf{q}_M^t\}$  para cada iteración  $t$ . Cada individuo constituye una solución potencial del problema a tratar, la cual se representa mediante una estructura de datos  $\mathbf{q}$ . Cada solución  $\mathbf{q}_j^t$  se evalúa utilizando una medida de su aptitud  $F$ . Luego se forma una nueva población (iteración  $t+1$ ) seleccionando los mejores individuos de  $\mathcal{S}^t$ . Algunos miembros de la nueva población sufren transformaciones unarias  $mu$  (mutación), que crean un nuevo individuo a partir de sí mismo ( $mu : \mathbf{q} \rightarrow \mathbf{q}$ ), y transformaciones de orden más alto  $rec$  (recombinación), que crean nuevos individuos combinando partes de algunos (dos o más) individuos ( $rec : \mathbf{q} \times \mathbf{q} \dots \times \mathbf{q} \rightarrow \mathbf{q}$ ). El algoritmo se ejecuta un cierto número de generaciones o hasta que se satisface algún criterio de detención. El mejor individuo se considera una solución del problema cercana a la óptima. El núcleo central de un AE lo constituyen los operadores de mutación y recombinación, siendo el diseño de los mismos dependiente del problema a tratar. El éxito de un AE depende en gran medida de que su diseño pueda captar las características esenciales del problema a resolver, esto es identificar y mantener los bloques constructivos de las mejores soluciones durante el proceso evolutivo.

Los AEDs constituyen un conjunto de métodos englobados dentro del paradigma de computación evolutiva. Si bien comparten características de los AEs, se distinguen porque las soluciones potenciales del problema, que forman la población, son consideradas como realizaciones de una variable aleatoria multidimensional. En los AEDs los operadores clásicos de cruzamiento y mutación se sustituyen por la estimación de la distribución de probabilidades subyacente de los miembros de la población y un posterior muestreo de dicha distribución.

Antes de comenzar a detallar los componentes fundamentales de un AED y los diferentes enfoques utilizados para su desarrollo, se realizará una breve descripción de la notación utilizada en este capítulo.

En este contexto, una propuesta de solución  $\mathbf{q}=(q_1, q_2, \dots, q_I)$  puede ser considerada como la realización de un vector aleatorio  $I$ -dimensional  $\mathbf{Q}=(Q_1, Q_2, \dots, Q_I)$ , donde cada  $Q_i$  representa una variable aleatoria de tipo discreto que puede asumir un conjunto finito de valores.

Asociado con  $\mathbf{Q}$  se tiene su función de distribución de probabilidades conjunta  $f_{\mathbf{Q}}$ , la cual permite evaluar la probabilidad de que  $\mathbf{Q}$  asuma el valor  $\mathbf{q}$ , ésto es  $\Pr(\mathbf{Q}=\mathbf{q})=f_{\mathbf{Q}}(\mathbf{q})=p(\mathbf{q})$ . Por otra parte, para cada variable aleatoria unidimensional  $Q_i$  queda definida su función de distribución marginal, ésto es,  $\Pr(Q_i=q_i)=p(q_i)=p_i$ . Las posibles dependencias entre las variables aleatorias intervinientes quedan expresadas a través de las probabilidades condicionales definidas como  $\Pr(Q_i=q_i/Q_I=q_I)=p(q_i/q_I)$ . Puede pensarse entonces que cada población de individuos en la iteración  $t$  del algoritmo, constituye una muestra aleatoria de tamaño  $M$  de la variable aleatoria  $\mathbf{Q}$ .

A diferencia de los AEs, donde los operadores específicos utilizan la información de los individuos de la población para guiar la búsqueda, en un AED se lleva a cabo el proceso de optimización a través de la construcción y evolución de un modelo probabilístico del espacio de soluciones. Las soluciones potenciales se evalúan a través de la función de evaluación o función de *fitness* y, realizando un proceso de selección de la o las mejores, se utiliza la información obtenida para actualizar el modelo probabilístico a partir del cual se muestrea la siguiente población. En la Figura 4.1 se muestra el esquema general de un AED, y el pseudocódigo correspondiente se presenta en la Figura 4.2 (Armañanzas y col., 2008).

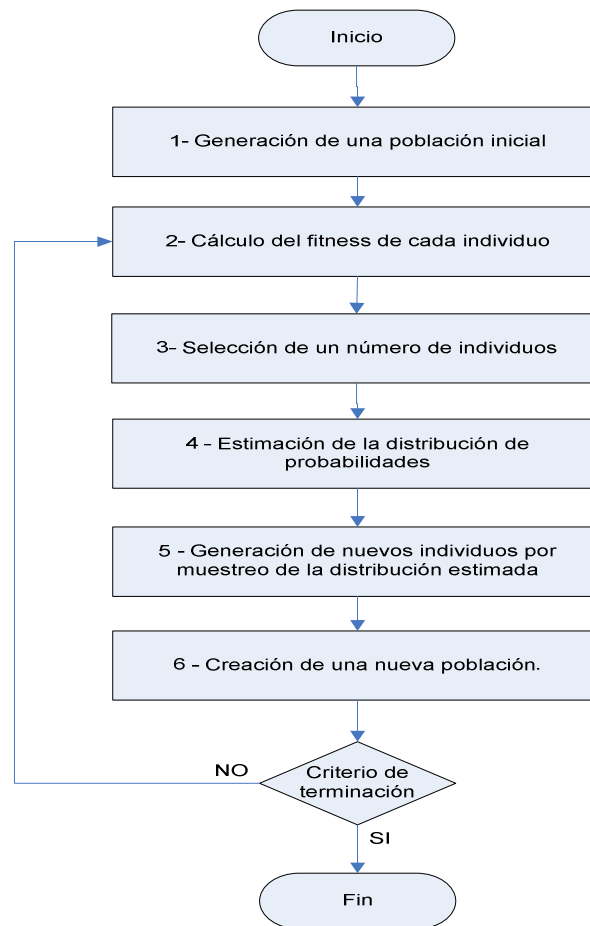


Figura 4.1: Esquema general de funcionamiento de un AED

$t \leftarrow 0$ . Generar  $M$  individuos aleatoriamente.

**repeat**

*Calcular la función de evaluación  $F$  de cada individuo*

*Seleccionar un conjunto  $\mathcal{D}_s^t$  de  $N \leq M$  puntos*

*Estimar un modelo probabilístico para  $\mathcal{D}^t$*

*Generar  $M$  nuevos individuos mediante muestreo desde la distribución estimada*

$t \leftarrow t + 1$

**until** *el criterio de terminación se alcance.*

Figura 4.2. Pseudocódigo para un AED

La capacidad de los AED de construir modelos que pueden captar las interrelaciones entre las variables y de este modo guiar la búsqueda hacia las regiones más prometedoras constituye la principal fortaleza de estos métodos para resolver

problemas de optimización complejos (Lozano y col., 2006). Por otra parte, el aprendizaje del modelo de distribución de probabilidades subyacente, ésto es la distribución conjunta de probabilidades, es la componente más importante y más compleja de los AED (Hauschild y Pelikan, 2011).

A continuación se describen brevemente los principales tipos de modelos probabilísticos que aparecen en la literatura, clasificados de acuerdo a la complejidad que exhiben al tratar de aprehender el grado de interdependencia entre las variables. En todos los casos se considerará, por razones de simplicidad y salvo una indicación en contrario, que las variables aleatorias discretas son de tipo binario.

#### 4.2.1 Modelos Univariados

A esta categoría pertenecen los modelos más sencillos de estimar. Ellos suponen que las variables aleatorias  $Q_i$  son independientes, y por lo tanto la distribución conjunta puede factorizarse como el producto de las marginales correspondientes. Si bien esto constituye una hipótesis simplificatoria de lo que ocurre en realidad en los problemas de optimización complejos, tiene como ventajas su fácil implementación y su bajo costo computacional, además de ofrecer posibilidades de escalamiento.

El algoritmo más simple dentro de esta categoría se denomina UMDA, *Univariate Marginal Distribution Algorithm*, (Mühlenbein, 1998). Para el caso de espacios de búsqueda en  $\{0,1\}^I$ , una población  $\mathcal{S}^t$  de tamaño  $M$  correspondiente a la iteración  $t$  puede pensarse como  $M$  realizaciones de  $I$  variables binomiales, donde el único parámetro a estimar es la probabilidad de éxito  $p_i$ ,  $i=1,\dots,I$ . Su estimador de máxima verosimilitud, basado en el conjunto seleccionado  $\mathcal{S}_s^{t-1}$  de tamaño  $N$ , está dado por:

$$p_i = P(Q_i = q_i) = p_i(q_i / \mathcal{D}_s^{t-1}) = \frac{\sum_{i=1}^N q_i}{N} \quad (4.1)$$

y la distribución conjunta queda expresada entonces como:

$$p(\mathbf{q}) = p_i(\mathbf{q} / \mathcal{D}_s^{t-1}) = \prod_{i=1}^I p_i \quad (4.2)$$

Con respecto al esquema utilizado en el algoritmo UMDA para generar el conjunto  $\mathcal{D}_s^t$  se han reportado esquemas basados en selección: proporcional, por torneo, y por truncamiento.

El algoritmo denominado cGA, *Compact Genetic Algorithm* (Harik y col., 1998), también pertenece a los enfoques que utilizan estadísticos de orden uno, pero difiere del anterior en el estimador empleado para actualizar el vector de probabilidades. En este caso, se efectúa una competencia entre dos individuos seleccionados previamente utilizando la función de evaluación. La competencia se lleva a cabo a nivel de cada variable unidimensional. Sea  $q_i^g$  la  $i$ -ésima posición el individuo con mejor función de evaluación y supóngase que toma un valor diferente del que tiene el individuo con peor valor. Si  $q_i^g = 1$ , entonces la  $i$ -ésima posición del vector de probabilidades se incrementa en un valor constante. Si, por el contrario,  $q_i^g = 0$  la correspondiente componente del vector de probabilidades disminuye en el mismo valor constante.

#### 4.2.2 Modelos Bivariados

Es sabido que uno de los aspectos fundamentales a tener en cuenta en los AGs es la adecuada creación, identificación y mantenimiento de los bloques constructivos, entendiendo por tales a las soluciones parciales del problema a resolver.

En general los mecanismos de cruzamiento propios de los AGs no preservan los bloques constructivos; las soluciones parciales se destruyen, lo que da como resultado la convergencia a subóptimos.

La capacidad de un AG de identificar los bloques constructivos y preservarlos bajo el operador de cruzamiento se conoce como “*linkage learning*”. Si bien el problema de la disrupción bajo cruzamiento no es compartido por los AEDs, la identificación y mantenimiento de los bloques constructivos es una característica deseable tanto en un AG como en los AEDs.

La construcción de un modelo probabilístico que contemple las interacciones entre las variables intervinientes constituye un enfoque alternativo para abordar el problema de “*linkage learning*”. En los AEDs las interacciones se representan mediante un modelo gráfico probabilístico, y se aprenden a partir de la identificación de las dependencias probabilísticas entre las variables.

Los modelos más sencillos que incorporan el aprendizaje estructural son aquellos que consideran sólo las posibles dependencias entre pares de variables. Algunos exponentes de este tipo de algoritmos se indican a continuación.

El algoritmo COMIT, *Combining Optimizers with Mutual Information Trees*, (Baluja y Davies, 1997) combina AED con optimización local. La estimación de la distribución de probabilidad de los individuos seleccionados en cada generación se hace mediante una red Bayesiana con estructura de árbol, utilizando en el aprendizaje el algoritmo *Maximum Weight Spanning Tree* propuesto por Chow y Liu (1968).

El algoritmo voraz denominado MIMIC, *Mutual Information Maximization for Input Clustering* (de Bonet y col., 1997), intenta encontrar la mejor factorización de la

distribución conjunta de probabilidades que sólo involucre probabilidades condicionales entre dos variables. La bondad de la distribución propuesta se mide utilizando la distancia de Kullback-Leiber.

El procedimiento llamado BMDA, *Bivariate Marginal Distribution Algorithm*, (Pelikan y Mühlenbein, 1999) utiliza una factorización de la distribución de probabilidad conjunta que sólo necesita estadísticos de orden dos.

### 4.2.3 Modelos Multivariados

En esta categoría se encuentran los AEDs que intentan contemplar, en principio, las dependencias entre todas las variables intervinientes en el problema. Por esta razón, son los modelos más complejos y en algunos casos el aprendizaje estructural requiere resolver un problema de optimización adicional.

El algoritmo FDA, *Factorized Distribution Algorithm*, (Mühlenbein y Mahning, 1999) es capaz de manejar interacciones de orden superior a dos a la vez que combina efectivamente las soluciones parciales. Se aplica a funciones aditivamente descomponibles utilizando sus conjuntos de definición para construir un grafo y a partir del mismo proponer una factorización de la distribución conjunta de probabilidades. Esta factorización permanece fija durante todas las iteraciones, actualizando en cada una de ellas sólo la estimación de las probabilidades. En este sentido FDA pertenece a la clase de AEDs que induce un aprendizaje paramétrico a partir de un modelo dado *a priori*. Esta información que debe ser conocida de antemano, y que no siempre está disponible para cualquier problema de optimización, constituye su principal desventaja.

Con el objetivo de generar una herramienta de optimización más general es que se proponen AEDs que puedan extender el aprendizaje paramétrico al aprendizaje estructural. En esta clase de algoritmos cabe mencionar las siguientes propuestas:

#### 4.2.3.1. AEDs que utilizan Modelos de Factorizaciones de Productos Marginales

La descomposición de la distribución conjunta de probabilidades de un vector aleatorio en un producto de múltiples distribuciones de probabilidades se denomina Factorización de Productos Marginales (MPM). Dicha factorización representa relaciones de independencia marginal entre las variables intervinientes, es decir las distribuciones multivariadas correspondientes a las factorizaciones están definidas sobre una partición del conjunto de todas las variables aleatorias consideradas. Por lo tanto, cada variable aleatoria aparecerá sólo en un factor de la factorización. Este tipo de modelo es aplicable cuando las variables del problema a resolver pueden agruparse en conjuntos mutuamente excluyentes.

Cabe mencionar que la MPM no utiliza distribuciones condicionales en la factorización, simplemente supone que todas las variables que aparecen en un factor se distribuyen de manera conjunta. Además el número de parámetros a estimar, esto es, el tamaño de las tablas de frecuencia asociadas a cada factor, crece de manera exponencial solamente con la cardinalidad de cada subconjunto de la partición propuesta. Estas características la diferencian notablemente de las redes Bayesianas, tal como se analizará en la siguiente subsección.

Dos ejemplos de AEDs que utilizan MPM son el UMDA y el ECGA, *Extended Compact Genetic Algorithm*, (Harik, 1999; Harik y col, 2006). El procedimiento UMDA corresponde al caso extremo de suponer que cada grupo de variables tiene sólo un elemento, lo que conduce a un AED que solamente considera estadísticos de orden uno. Por otra parte, en el procedimiento ECGA se utilizan particiones del conjunto de variables con cardinalidad diferente; se dividen las variables en grupos (*clusters*) independientes y cada uno de estos grupos se asume como una variable simple. La construcción del modelo (estructura del modelo probabilístico) se lleva a cabo



iterativamente mediante un algoritmo voraz, que comienza suponiendo que todas las variables son independientes. En cada iteración se unen dos clusters para mejorar la calidad del modelo propuesto, medida en términos de una métrica compuesta, denominada MDL, *Minimum Description Length Metric*. La complejidad combinada tiene en cuenta dos factores: i) la entropía de las distribuciones marginales asociadas a la función Complejidad de la Población Comprimida; ii) la dimensión del modelo relacionada con la función Complejidad del Modelo, que tiene en cuenta el número de parámetros a estimar necesarios para especificar la distribución marginal de cada grupo.

Una vez que la etapa de aprendizaje estructural se completa, a partir de un conjunto de individuos seleccionados se calculan los valores asociados a las distribuciones de probabilidades conjuntas resultantes de la partición propuesta. Posteriormente se realiza el muestreo correspondiente para generar la próxima población. En la Figura 4.3 se muestra un pseudocódigo para ECGA.

```
t ← 0. Generar M individuos aleatoriamente.  
repeat  
    Calcular la función de evaluación F para cada individuo de la población  
    Seleccionar por torneo un conjunto  $\mathcal{S}'_s$  de  $N \leq M$  individuos  
    Modelar la población usando una búsqueda voraz  
    Si el modelo converge, detener la búsqueda  
    Generar una nueva población usando el modelo construido  
    t ← t + 1  
until el criterio de terminación se alcance.
```

Figura 4.3. Pseudocódigo para ECGA

Otra propuesta de un AEDs similar a ECGA es el AffEDA, *Affinity Propagation Estimation Distribution Algorithm*, (Santana y col., 2010). Éste comienza generando en primera instancia una matriz de información mutua (**IM**) entre todas las variables. Dadas

dos variables aleatorias  $X$  e  $Y$ , se define la información mutua,  $IM$ , entre ambas mediante la siguiente ecuación:

$$IM(X, Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log_2 \left( \frac{p(x, y)}{p(x)p(y)} \right) \quad (4.3)$$

donde:  $p(x, y)$  es el valor de la función de distribución conjunta para el vector aleatorio  $(X, Y)$ , esto es la probabilidad de que ocurra el evento  $(X=x, Y=y)$ . Por otra parte,  $p(x)$  y  $p(y)$  representan los valores de las distribuciones marginales para las variables  $X$  e  $Y$  respectivamente.

Si las variables aleatorias son independientes, entonces la distribución conjunta puede ser factorizada como el producto de las distribuciones marginales,  $p(x, y) = p(x)p(y)$  y el valor de la información mutua entre  $X$  e  $Y$  es cero.

La partición del conjunto de variables en subconjuntos mutuamente excluyentes se lleva a cabo mediante el algoritmo AP, *Affinity Propagation*, (Frey y Dueck, 2007) que es un procedimiento de agrupamiento (*clustering*). Dado un conjunto de puntos y una medida de similitud entre ellos, la técnica proporciona grupos de puntos similares, y además para cada grupo da un ejemplar representativo. La medida de similaridad es la  $IM$  entre cada par de variables aleatorias, la cual expresa la información que ellas comparten. El procedimiento de AP está basado en el intercambio de mensajes entre puntos (potenciales soluciones) hasta que la condición de convergencia sea alcanzada. Los mensajes son de dos tipos, responsabilidad y disponibilidad, y son intercambiados entre los puntos indicando la afinidad que cada uno de ellos tiene respecto de otro para actuar como su ejemplar. El mensaje de responsabilidad se envía desde un punto a los candidatos a ejemplares y refleja cuan bien adaptado está el receptor para representar al emisor del mensaje. Los mensajes de disponibilidad, a su vez, se emiten desde un

candidato a ejemplar con destino a un punto y posee información de cuan apropiado puede ser el emisor para ser el ejemplar del receptor. Los puntos reciben mensajes de disponibilidad de varios candidatos a ejemplares, a la vez que éstos reciben mensajes de responsabilidad de varios puntos. Una vez que el algoritmo converge, entrega la estructura de la partición del conjunto de variables.

El número de grupos no se determina de antemano, sino que es un resultado de la aplicación del procedimiento AP. El tamaño de cada grupo fija el número de factores que aparecen en cada distribución marginal, y por lo tanto el número de probabilidades a estimar. Para el caso de variables binarias, como el PDRS, si el número de factores del grupo es  $\lambda$ , entonces es necesario estimar  $2^\lambda$  parámetros. Dado que la población de soluciones con que se cuenta en una iteración debería proveer una buena estimación del conjunto de parámetros para el factor más grande, se incorpora al algoritmo de agrupamiento una restricción acerca del tamaño máximo de los grupos. Este no debe superar el valor  $\lambda_{\max} = \log_2 N$ , siendo  $N$  el número de individuos de la población seleccionada.

Una vez que el modelo de grupos independientes se generó, es decir, concluyó la fase de aprendizaje de la estructura del modelo, se calculan los parámetros del mismo (tablas de frecuencia) y se muestrea de manera similar a como se realiza usando el ECGA. En la Figura 4.4 se muestra un pseudocódigo de AffEDA.

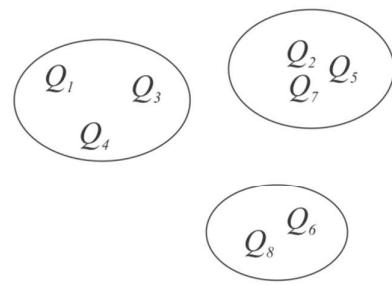
```

t ← 0. Generar M individuos aleatoriamente.
repeat
    Calcular la función de evaluación F para cada individuo de la población
    Seleccionar por torneo un conjunto S' de N ≤ M individuos
    Computar la IM entre cada par de variables
    Aplicar algoritmo AP para obtener una MPM de tamaño restringido
    Generar una nueva población usando el modelo construido
    t ← t + 1
until el criterio de terminación se alcance.
    
```

Figura 4.4. Pseudocódigo para AffEDA

Con el objetivo de ilustrar esta metodología, se va a considerar una planta constituida por 8 corrientes y 5 unidades de proceso. En este caso, el vector aleatorio **Q** es un vector de dimensión ocho. La Figura 4.5 muestra, para una población seleccionada de 11 individuos, los grupos obtenidos luego de la aplicación del algoritmo AP. Se han formado tres grupos definidos por los siguientes conjuntos de variables: { $Q_1, Q_3, Q_4$ }, { $Q_2, Q_5, Q_7$ } y { $Q_6, Q_8$ }. En dicha tabla #ind representa la cardinalidad del individuo.

# ind	$Q_1$	$Q_2$	$Q_3$	$Q_4$	$Q_5$	$Q_6$	$Q_7$	$Q_8$	F
1	1	0	0	0	0	1	0	0	3500
2	1	0	0	0	0	1	0	0	3500
3	1	0	0	1	0	0	1	0	3500
4	1	1	0	0	0	0	0	1	4000
5	1	0	0	0	0	1	0	0	3500
6	0	1	0	0	0	1	0	0	4000
7	1	0	0	0	0	0	0	1	2500
8	1	1	0	0	0	0	0	1	4000
9	1	0	0	0	0	1	0	0	3500
10	1	1	0	0	0	0	0	1	4000
11	1	1	0	0	0	0	0	1	4000



(a)

(b)

Figura 4.5. Ejemplo de grupos obtenidos empleando la técnica AP

La función de distribución conjunta de este modelo estructural se formula como sigue:

$$p(\mathbf{q}) = p(q_1, q_3, q_4)p(q_2, q_5, q_7)p(q_6, q_8) \tag{4.4}$$

El aprendizaje de los parámetros del modelo está basado en las tablas de frecuencia que se muestran en la Figura 4.6.

Grupo 1						Grupo 2									
#ind	$Q_1$	$Q_3$	$Q_4$	Valores		Prob	#ind	$Q_2$	$Q_5$	$Q_7$	Valores		Prob		
1	1	1	1	0	0	0	0,0909	1	1	1	1	0	0	0	0,4545
2	1	0	0	0	0	1	0,0000	2	1	0	0	0	0	1	0,0909
3	1	1	0	0	1	0	0,0000	3	1	1	0	0	1	0	0,0000
4	0	0	0	0	1	1	0,0000	4	0	0	0	0	1	1	0,0000
5	1	0	0	1	0	0	0,2727	5	1	0	0	0	0	0	0,4545
6	1	0	1	1	0	1	0,1818	6	1	0	1	0	0	1	0,0000
7	1	0	0	1	1	0	0,2727	7	1	0	0	0	1	0	0,0000
8	1	1	0	1	1	1	0,1818	8	1	1	0	1	1	1	0,0000
9	1	1	0					9	1	1	0				
10	1	1	1					10	1	1	1				
11	1	0	1					11	1	0	1				

Grupo 3					
#ind	$Q_6$	$Q_8$	Valores		Prob
1	1	0	0	0	0,0909
2	1	0	0	1	0,4545
3	0	0	1	0	0,4545
4	0	1	1	1	0,0000
5	1	0			
6	1	0			
7	0	1			
8	0	1			
9	1	0			
10	0	1			
11	0	1			

Figura 4.6. Ejemplo de estimación de parámetros para AffEDA

Por ejemplo, la combinación de valores  $Q_1=1$ ,  $Q_3=0$  y  $Q_4=0$  para el primer grupo se encuentra tres veces en la población seleccionada (individuos 2, 5 y 7). Por lo tanto la

probabilidad de ocurrencia de este evento (*Valores Prob*) es  $3/11=0.2727$ . Este procedimiento se lleva a cabo para los todos posibles eventos en cada grupo. Finalmente, se calcula el vector de probabilidades acumulado a partir del cual se muestrea una nueva población.

#### 4.2.3.2. AEDs que utilizan Modelos de Redes Bayesianas

Los modelos presentados en el punto anterior presuponen independencia entre conjuntos de variables, en otras palabras, la dependencia está restringida a las variables que pertenecen a un mismo agrupamiento. Cuando el conjunto de dependencias es más complejo, las Redes Bayesianas, RB, son una herramienta más adecuada, ya que permiten hallar una factorización de la función de distribución de probabilidades conjunta. De forma general, una RB, está constituida por un grafo acíclico dirigido, el cual representa el conjunto de dependencias entre las variables aleatorias, o sea el modelo estructural, y un conjunto de parámetros, las distribuciones de probabilidades condicionales. En la literatura se encuentran diferentes AEDs que utilizan RB como modelo gráfico probabilístico para la estimación de la distribución conjunta. Entre ellos cabe mencionar los métodos presentados a continuación.

El algoritmo BOA, *Bayesian Optimization Algorithm*, (Pelikan y col., 1999) se inicializa con una población de soluciones candidatas, y en cada iteración se construye el modelo probabilístico que mejor ajuste los datos observados partiendo del conjunto de las mejores soluciones previamente encontradas. La métrica utilizada para medir la bondad del ajuste es la BDE, *Bayesian Dirichlet Equivalence*, (Heckerman y col., 1995), y la técnica de búsqueda de la mejor estructura es una heurística de tipo voraz. La cardinalidad del espacio de búsqueda se acota de manera tal que cada variable no puede contar con más de  $l$  padres. La Figura 4.7 muestra un esquema general para el algoritmo BOA.

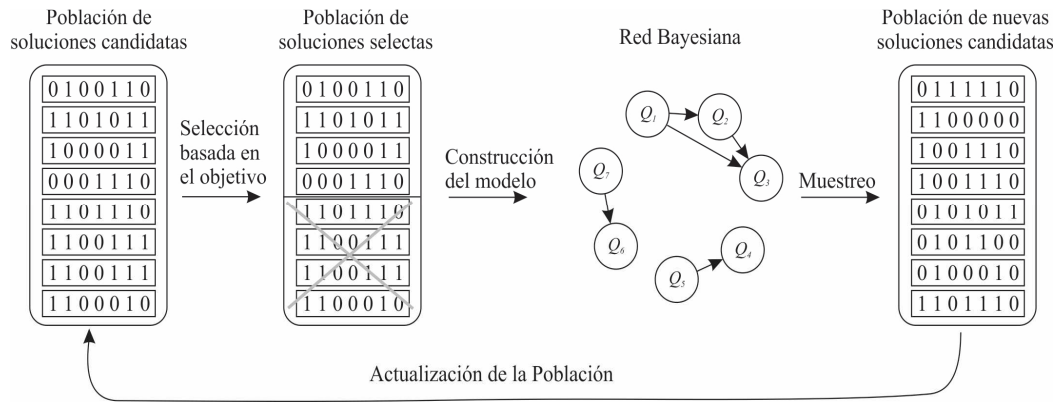


Figura 4.7. Esquema general para el algoritmo BOA

Una modificación más reciente de este algoritmo es el iBOA, *Incremental Bayesian Optimization Algorithm*, (Pelikan y col., 2008). Este propone remover la población de soluciones potenciales mantenida por BOA, y utilizar un mecanismo de actualización incremental para el aprendizaje de la RB y la estimación de los parámetros.

Otra técnica multivariada que utiliza RB para codificar la factorización de la distribución es el algoritmo EBNA, *Estimation of Bayesian Network Algorithm*, (Etxeberria y Larrañaga, 1999), y sus diferentes variantes EBNA<sub>BIC</sub>, EBNA<sub>K2+pen</sub>, EBNA<sub>PC</sub>. Éstos se diferencian en el método utilizado para efectuar el aprendizaje estructural de la RB. Los dos primeros aplican técnicas de búsqueda y calificación (*score*) para tal fin, mientras que el último emplea test de hipótesis de independencia estadística (Larrañaga y col., 2012).

A modo de ejemplo se muestra en la Figura 4.8, para el mismo proceso de 8 corrientes y 5 equipos mencionado anteriormente, el modelo estructural proporcionado por EBNA, para la población seleccionada representada en la Figura 4.5.

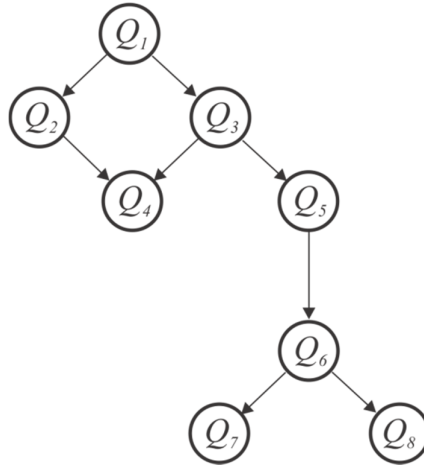


Figura 4.8. Red Bayesiana a partir de la población seleccionada

Para la construcción del modelo se utilizó el algoritmo goloso B (Buntine, 1991), que comienza con un estructura sin arcos e intenta en cada iteración agregar arcos que ofrezcan la máxima mejora en la medida de ajuste BIC, *Bayesian Information Metric*, (Schwarz, 1978). El algoritmo se detiene cuando la incorporación de nuevos enlaces no produce un incremento de la métrica utilizada. El modelo obtenido representa las dependencias condicionales entre las variables intervinientes y su conjunto de parámetros son estimados aplicando el método de máxima verosimilitud. La función de distribución conjunta inferida a partir de la red propuesta es la siguiente:

$$p(\mathbf{q}) = p(q_1)p(q_2/q_1)p(q_3/q_1)p(q_4/q_2, q_3)p(q_5/q_3)p(q_6/q_5)p(q_7/q_6)p(q_8/q_6) \quad (4.5)$$

La factorización encontrada tiene un número de factores significativamente más grande que la expresión obtenida cuando se utiliza el enfoque MPM. Esto se traduce en un importante aumento tanto de la cantidad de parámetros a estimar como del esfuerzo computacional. En la Figura 4.9 se ejemplifica el cálculo de la probabilidad condicional  $p(q_2/q_1)$ .



# ind	$Q_1$	$Q_2$	Probabilidad Condicional
1	1	0	$p(Q_2 = 0 / Q_1 = 0) = \frac{p(Q_2 = 0, Q_1 = 0)}{p(Q_1 = 0)} = \frac{0/11}{1/11} = 0$
2	1	0	
3	1	0	
4	1	1	$p(Q_2 = 0 / Q_1 = 1) = \frac{p(Q_2 = 0, Q_1 = 1)}{p(Q_1 = 1)} = \frac{6/11}{10/11} = 0.6$
5	1	0	
6	0	1	
7	1	0	$p(Q_2 = 1 / Q_1 = 0) = \frac{p(Q_2 = 1, Q_1 = 0)}{p(Q_1 = 0)} = \frac{1/11}{1/11} = 1$
8	1	1	
9	1	0	
10	1	1	$p(Q_2 = 1 / Q_1 = 1) = \frac{p(Q_2 = 1, Q_1 = 1)}{p(Q_1 = 1)} = \frac{4/11}{10/11} = 0.4$
11	1	1	

Figura 4.9. Estimación de parámetros: Evaluación de las probabilidades condicionales

### 4.3 Resultados

En esta sección se presentan los resultados de la aplicación de AEDs para la resolución de los PDRS, denominados Diseño 1 y Diseño 2, presentados en el Capítulo 3 de esta tesis. Se realizaron experiencias implementando un AED de orden uno y dos tipos diferentes de AEDs con aprendizaje estructural. Teniendo en cuenta que en el PDRS existen naturalmente dependencias entre las variables, se trata de investigar el impacto de los diferentes modelos probabilísticos sobre las soluciones obtenidas.

En primer lugar se analizó el desempeño del algoritmo UMDA. Éste supone que todas las variables son independientes entre sí, por lo tanto se forman grupos de un solo elemento. Luego se consideró una extensión natural de esta propuesta, el algoritmo AffEDA, el cual utiliza un MPM y asume que todas las variables que aparecen en un grupo se distribuyen de manera conjunta. Por último se implementó el algoritmo EBNA, que expresa las dependencias entre las variables en términos de probabilidades

condicionales, a diferencia del AffEDA. Las dos últimas técnicas se ejecutaron utilizando las funciones para MatLab provistas por MATEDA (Santana y col., 2009).

Las características y valores de los parámetros comunes a los algoritmos implementados son:

- a) Población inicial: se genera una población inicial de potenciales soluciones,  $\mathcal{D}^0$ , de manera aleatoria.
- b) Tamaño de la población: las experiencias se realizan con una población de  $M=50$  individuos.
- c) Método de Selección: se emplea el método de selección por torneo. Se calcula el valor de la función  $F$  para los individuos de la población y se forma por competencia un conjunto seleccionado de  $N=25$ .
- d) Función de evaluación: se utiliza la descripta por la ecuación 3.13.
- e) Criterio de detención: se emplea el número máximo de generaciones,  $maxiter=100$ .

#### 4.3.1 Resultados de la aplicación del Algoritmo UMDA

Para el Diseño 1, los resultados obtenidos son comparables a los alcanzados empleando Búsqueda Dispersa (BD), siendo el valor de los estadísticos media, desvío estándar y coeficiente de variación correspondientes a BD levemente menores. En el escenario del Diseño 2, en cambio, el desempeño del algoritmo UMDA es más pobre. Los resultados de la optimización para 100 experiencias independientes se muestran en las Tablas 4.1 y 4.2.

Mínimo	Media	Desvío Estándar	CV(%)
533.56	553.76	48.31	8.7

Tabla 4.1. Estadísticos de la solución para el Diseño 1 – UMDA

Mínimo	Media	Desvío Estándar	CV(%)
894.86	1261.75	453.47	36.1

Tabla 4.2. Estadísticos de la solución para el Diseño 2 – UMDA

### 4.3.2 Resultados de la aplicación de Algoritmos con aprendizaje estructural

A continuación se describe el comportamiento del algoritmo AffEDA en la resolución de los diseños propuestos. Los resultados de la optimización se muestran en las Tablas 4.3 y 4.4. Los mismos evidencian un muy buen desempeño para el Diseño 1, con un coeficiente de variación un 50% inferior, 4.7 %, que el correspondiente al algoritmo UMDA. Sin embargo, el incremento en el esfuerzo computacional ocasionado al complejizar el modelo de distribución de probabilidades, no origina el mismo beneficio para el Diseño 2.

Mínimo	Media	Desvío Estándar	CV(%)
533.56	538.7	24.11	4.7

Tabla 4.3. Estadísticos de la solución para el Diseño 1 – AffEDA

Mínimo	Media	Desvío Estándar	CV(%)
894.86	1258.42	452.94	35.9

Tabla 4.4. Estadísticos de la solución para el Diseño 2 – AffEDA

Respecto del algoritmo EBNA los valores obtenidos luego de la optimización se presentan en las Tablas 4.5 y 4.6, en las cuales puede apreciarse que el desempeño del

algoritmo EBNA es inferior al de AffEDA en todos los casos. Para el Diseño 2, el valor mínimo indicado por EBNA es superior al que proporciona cualquier otro método. Sin embargo, cabe mencionar que el tamaño de la población es uno de los parámetros que influyen en el desempeño de las técnicas basadas en modelos de RB. Por lo tanto, si la cantidad de información disponible es mayor, esto es, si el tamaño de la población se incrementa, debería mejorar la bondad de la inferencia estadística basada en la estructura y parámetros del modelo probabilístico. El impacto de este factor sobre la calidad de las soluciones se analizará en más detalle en el Capítulo 6.

Mínimo	Media	Desvío Estándar	CV(%)
542.36	681.02	72.54	10.6

Tabla 4.5. Estadísticos de la solución para el Diseño 1 – EBNA

Mínimo	Media	Desvío Estándar	CV(%)
915.06	1555.77	373.57	24

Tabla 4.6. Estadísticos de la solución para el Diseño 2 – EBNA

#### 4.4 Conclusiones

En este capítulo se presentó y analizó la metodología de resolución de problemas de optimización combinatoriales basada en AEDs. Estos se incluyen dentro del paradigma de la computación evolutiva, ya que comparten algunas características con, por ejemplo, los AGs. Sin embargo, el núcleo central en los AEDs lo constituye la estimación y actualización del modelo probabilístico que mejor represente a una población observada de potenciales soluciones.

Se describieron los principales enfoques encontrados en la literatura en orden creciente de complejidad respecto del modelo de distribución de probabilidades propuesto. Se consideraron en primer lugar aquellos algoritmos que suponen independencia entre las variables intervinientes, y que por lo tanto, sólo requieren la estimación y actualización de las distribuciones marginales de probabilidades. Por último se analizaron los AEDs que utilizan tanto el aprendizaje estructural, para reflejar las dependencias entre el conjunto de variables aleatorias, como el aprendizaje paramétrico.

Las metodologías de resolución presentadas se aplicaron a dos PDRS. Del análisis de resultados surge que tanto el algoritmo que supone independencia entre las variables como las técnicas más complejas resuelven correctamente el Diseño 1. El inconveniente surge nuevamente con el Diseño 2, el cual a pesar de involucrar el mismo número de variables binarias que el Diseño 1, presenta un grado de dificultad diferente.

La complejidad adicional que implica considerar un modelo gráfico probabilístico de mayor orden, ya sea por los mayores tiempos de cómputo o por el hecho de sumar un problema de optimización adicional como en el caso de aprendizaje de RB, no se traduce en una mejora sustancial en la calidad y repetitividad de las mejores soluciones obtenidas para el Diseño 2. Cabe mencionar que los AEDs utilizados, hasta este punto, para resolver el PDRS no tienen incorporado ningún conocimiento específico del problema, parten de poblaciones inicializadas al azar y no están hibridizados con ningún mecanismo de búsqueda local. Aún así evidencian un desempeño aceptable ya que son capaces de hallar buenas soluciones.

## APÉNDICE DEL CAPÍTULO 4

### Modelos Gráficos Probabilísticos

Los modelos gráficos probabilísticos (Howard y Matheson, 1981; Lauritzen, 1996; Pearl, 1988) son herramientas utilizadas para representar interacciones en sistemas complejos en términos de dependencias probabilísticas. De manera general constan de dos elementos: una estructura que refleja las dependencias entre variables y una colección de valores numéricos, parámetros, que usualmente son probabilidades condicionales que reflejan el grado de las interacciones. Un modelo gráfico probabilístico se representa mediante un grafo  $G$ , en el que cada nodo representa una variable aleatoria y los arcos entre nodos indican las dependencias condicionales entre las variables. Si la estructura de la red es un grafo acíclico dirigido, éste representa una factorización de la distribución conjunta  $p(\mathbf{q})$ , para el vector aleatorio  $\mathbf{Q}=(Q_1, \dots, Q_I)$ . Este tipo de modelo es conocido como RB.

Dado un grafo acíclico dirigido  $G$  para el cual cada nodo representa una variable  $Q_i$  y  $\mathbf{Pa}_i$  es el conjunto de padres de la  $i$ -ésima variable (conjunto de nodos desde los cuales sale un arco hacia el  $i$ -ésimo nodo), entonces la distribución conjunta inducida por la estructura gráfica está dada por:

$$p(\mathbf{q}) = \prod_{i=1}^I p(q_i / \mathbf{pa}_i) \quad (\text{A4.1})$$

donde  $p_{ai}$  representa un valor asumido por la variable  $\mathbf{Pa}_i$  de su posible conjunto de  $s_i$  instancias.

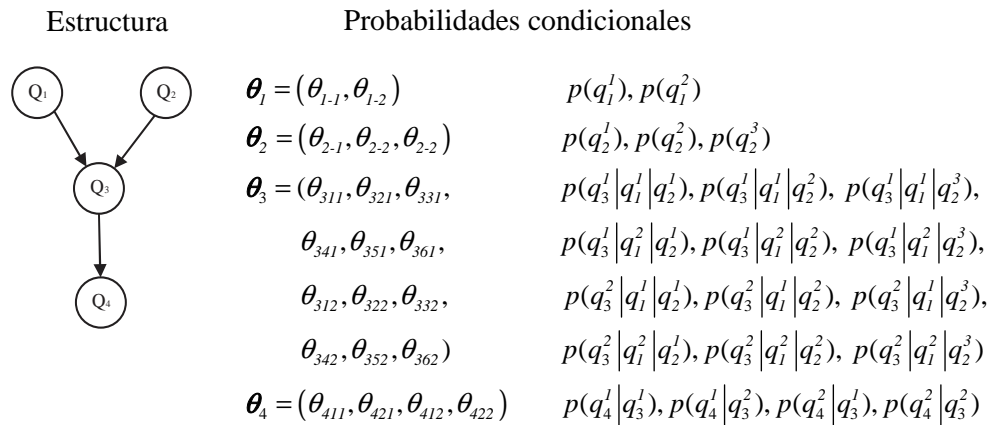
Se supone además que la función  $p(\mathbf{q})$  depende de un conjunto finito de parámetros  $\Theta_G = (\theta_1, \dots, \theta_l)$  que es necesario estimar, por lo que la ecuación anterior queda:

$$p(\mathbf{q} / \Theta_G) = \prod_{i=1}^l p(q_i / \mathbf{pa}_i, \theta_i) \tag{A4.2}$$

Si la variable aleatoria  $Q_i$  puede asumir  $r_i$  posibles valores, la probabilidad condicional de que ésta asuma el valor  $k$ , dado que su conjunto de padres está en el  $j$ -ésimo valor es:

$$p(q_i^k / \mathbf{pa}_i^j, \theta_i) = \theta_{ijk}, \quad \text{con } \theta_{ijk} > 0 \tag{A4.3}$$

El modelo gráfico probabilístico queda entonces definido por un par  $(G, \Theta)$  en el cual se incluyen la componente estructural del modelo y el conjunto de parámetros asociados. En la Figura A4.1 se muestra un esquema ilustrativo, para el caso de cuatro variables aleatorias, donde  $Q_1$ ,  $Q_3$  y  $Q_4$  pueden asumir dos valores y  $Q_2$  tiene tres posibles valores (Miquelez Echegaray, 2010).



Factorización de la probabilidad conjunta

$$p(q_1, q_2, q_3, q_4) = p(q_1) \cdot p(q_2) \cdot p(q_3 | q_1, q_2) \cdot p(q_4 | q_3)$$

Figura A4.1. Esquema básico de un modelo gráfico probabilístico

La implementación de RB en AEDs involucra el aprendizaje de la estructura, la determinación del conjunto de parámetros y la posterior simulación del modelo aprendido. Los diferentes enfoques encontrados en la literatura acerca de esta metodología en AEDs difieren esencialmente en la forma de realizar algunos de estos pasos. Cabe mencionar que el aprendizaje de una RB está catalogado como un problema NP-duro, y por lo tanto la utilización de algoritmos exactos para su resolución está restringida a casos que involucren un número reducido de variables.

Las etapas de la implementación de RB en AEDs se describen a continuación:

- a) Aprendizaje de la estructura de una RB. Consta de dos componentes básicos: una métrica para evaluar la bondad del ajuste de la red propuesta a los datos y un método de búsqueda de redes posibles con el mayor valor de ajuste.
- b) Respecto de las métricas utilizadas para evaluar la bondad de ajuste pueden mencionarse las siguientes: BD, *Bayesian Dirichlet*, (Hekermann y col., 1995) y BIC, *Bayesian Information Criterion*, (Schwarz, 1978).
- c) Debido a la complejidad computacional inherente al problema de aprendizaje en la fase de búsqueda, normalmente se utilizan algoritmos heurísticos, tales como búsqueda local, algoritmos voraces, etc. Dentro de éstos cabe mencionar al algoritmo goloso B (Buntine, 1991), que comienza con un estructura sin arcos y que intenta en cada iteración agregar arcos que ofrezcan la máxima mejora en la medida de ajuste utilizada. El algoritmo se detiene cuando la incorporación de nuevos enlaces no produce un incremento de la métrica utilizada. En algunos casos, la dimensión del espacio de búsqueda de posibles redes se acota permitiendo a lo sumo que  $k$  enlaces ingresen a un nodo. Esta restricción intenta reducir el costo computacional asociado tanto a esta etapa como a la de



generación de nuevas instancias de la red, a la vez que restringe el orden de interacciones a modelar.

- d) Determinación del conjunto de parámetros. La estimación de las distribuciones marginales y/o condicionales que aparecen en la factorización de la distribución conjunta de probabilidades se lleva a cabo utilizando el método de máxima verosimilitud con corrección de Laplace.
- e) Simulación de la red aprendida. Uno de los algoritmos más utilizados para generar nuevas instancias del conjunto de variables es el denominado *Probabilistic Logic Sampling*, PLS, (Henrion, 1998). Las variables se ordenan de modo tal que las asignaciones se realizan primero sobre los nodos padres,  $\mathbf{Pa}_i$ , y luego sobre la variable  $Q_i$ , de acuerdo a las probabilidades estimadas,  $p(q_i/\mathbf{pa}_i)$ . Un ordenamiento con estas características se denomina ordenamiento ancestral.

## 4.5 Nomenclatura

$\mathcal{D}$	Población de individuos
$\mathcal{D}^t$	Población de individuos en la iteración $t$
$\mathcal{D}_s^t$	Población de individuos seleccionada en la iteración $t$
$F$	Función de evaluación o función de <i>fitness</i>
$f_Q$	Función de distribución de probabilidad conjunta
$G$	Grafo acíclico dirigido

---

<b>IM</b>	Matriz de información mutua
$p$	Asignación de probabilidades
$p(q_i/q_j)$	Probabilidad condicional de $q_i$ dado $q_j$
<b><math>Pa_i</math></b>	Conjunto de padres para la $i$ -ésima variable
$q$	Realización de la variable $Q$
$Q$	Variable aleatoria unidimensional
<b>q</b>	Realización del vector aleatorio <b>Q</b>
<b>Q</b>	Vector de variables aleatorias
Letras Griegas	
$\lambda$	Número de factores en la factorización de la distribución conjunta
$\Theta$	Conjunto de parámetros
$\theta_i$	Vector de parámetros asociado a la $i$ -ésima variable

#### 4.6 Acrónimos

AE	Algoritmo Evolutivo
AED	Algoritmo de Estimación de Distribuciones
AffEDA	<i>Affinity Propagation Estimation Distribution Algorithm</i>
AG	Algoritmo Genético
AP	<i>Affinity Propagation</i>

---

B	Algoritmo de Buntine
BDE	<i>Bayesian Dirichlet Equivalence</i>
BIC	<i>Bayesian Information Metric</i>
BMDA	<i>Bivariate Marginal Distribution Algorithm</i>
BOA	<i>Bayesian Optimization Algorithm</i>
cGA	<i>Compact Genetic Algorithm</i>
COMIT	<i>Combining Optimizers with Mutual Information Trees</i>
EBNA	<i>Estimation of Bayesian Network Algorithm</i>
ECGA	<i>Extended Compact Genetic Algorithm</i>
FDA	<i>Factorized Distribution Algorithm</i>
iBOA	<i>Incremental Bayesian Optimization Algorithm</i>
<i>IM</i>	Información mutua
MDL	<i>Minimum Description Length Metric</i>
MIMIC	<i>Mutual Information Maximization for Input Clustering</i>
MPM	Factorización de Productos Marginales
PDRS	Problema de Diseño de Redes de Sensores
PLS	<i>Probabilistic Logic Sampling</i>
RB	Redes Bayesianas

UMDA      *Univariate Marginal Distribution Algorithm*

#### 4.7 Referencias

- Armañanzas, R., Inza, I., Santana, R., Saeys, Y., Flores, J., Lozano, J. A review of estimation of distribution algorithms in bioinformatics, *BioData Mining*, **1**, (2008).
- Baluja, S., Davies, S. Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space. *Technical report, Carnegie Mellon Report*, CMU-CS-97-107, (1997).
- Buntine, W. Theory refinement on Bayesian networks. *Proc. 7th Conf. Uncertainty Artif. Intell.*, 52–60, (1991).
- Chow, C., Liu, C. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, **14**(3), 462-467, (1968).
- de Bonet, J. S., Isbell, C. L., y Viola, P. MIMIC: Finding optima by estimating probability densities. *Advances in Neural Information Processing Systems*, **9**, 424-431, Cambridge, MA. M. Mozer, M. Jordan and Th. Petsche eds, (1997).
- Etxeberria, R., Larrañaga, P. Global optimization with Bayesian networks. *Special Session on Distributions and Evolutionary Optimization*, 332-339. La Habana, Cuba. II Symposium on Artificial Intelligence, CIMAF99, (1999).
- Frey, B. J., Dueck, D. Clustering by passing messages between data points. *Science*, **315**, 972–976, (2007).
- Harik, G. Linkage learning via probabilistic modeling in the ECGA. *IlligAL Report No. 99010*, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL, (1999).
- Harik, G., Lobo, F. G., Goldberg, D. E. The compact genetic algorithm. *Proceedings of the IEEE Conference on Evolutionary Computation*, 523-528, Piscataway, NJ, (1998).

- Harik, G.R.; Lobo, F.; Sastry, K. Linkage Learning via Probabilistic Modeling in the Extended Compact Genetic Algorithm (ECGA). *Studies in Computational Intelligence*, **33**, 39–61, (2006).
- Hauschild, M., Pelikan, M. An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation*, **1**, 111–128, (2011).
- Heckerman, D., Geiger, D., y Chickering, D. M. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, **20**, 197-243, (1995).
- Henrion, M. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. *Uncertainty in Artificial Intelligence*, **2**, 149-163. J. F. Lemmer and L. N. Kanal eds., North-Holland, Amsterdam, (1998).
- Howard, R., Matheson, J. Influence diagrams. *Readings on the Principles and Applications of Decision Analysis*, **2**, 721-764. Howard, R. y Matheson, J., editors. Strategic Decision Group, Menlo Park, California, (1981).
- Larrañaga, P., Hossein, K., Bielza, C., Santana, R. A review on probabilistic graphical models in evolutionary computation. *J Heuristics* **18**, 795–819, (2012).
- Lauritzen, S. L. *Graphical Models*. Oxford University Press, (1996).
- Lozano, J., Larrañaga, P., Inza, I., Bengoetxea, E. *Edas Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*. Springer-Verlag, (2006).
- Miquélez Echegaray, T. *Avances en Algoritmos de Estimación de Distribuciones. Alternativas en el Aprendizaje y Representación de Problemas*. Tesis Doctoral, Universidad del País Vasco, (2010).
- Mühlenbein, H. The equation for response to selection and its use for prediction. *Evolutionary Computation*, **5**, 303–346, (1998).
- Mühlenbein, H., Mahning, T. FDA - a scalable evolutionary algorithm for the optimization of additively decomposed functions. *Evolutionary Computation*, **7**(4), 353-376, (1999).

- Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, California, (1988).
- Pelikan, M., Mühlenbein, H. The bivariate marginal distribution algorithm. , *Advances in Soft Computing Engineering Design and Manufacturing*, 521-535, Roy, R., Furuhashi, T., y Chandhory, P. K., editors London. Springer-Verlag, (1999).
- Pelikan, M., Goldberg, D. E., Cantú-Paz, E. BOA: The Bayesian optimization algorithm. *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, Orlando FL, **1**, 525-532, San Francisco, California. Morgan, Kaufmann Publishers, (1999).
- Pelikan, M., Sastry, K., Goldberg, D. iBOA: The Incremental Bayesian Optimization Algorithm. *MEDAL Report No. 2008002*, (2008).
- Santana, R., Echegoyen, C., Mendiburu, A., Bielza, C., Lozano, J., Larrañaga, P., Armañanzas, R., Shakya, S. MATEDA: A suite of EDA programs in Matlab. *Research Report EHU-KZAA-IK-2/09*, Department of Computer Science and Artificial Intelligence, Universidad del País Vasco, (2009).
- Santana, R.; Larrañaga, P.; Lozano, J. Learning Factorizations in Estimation of Distribution Algorithms Using Affinity Propagation. *Evolutionary Computation* **18**(4), 515–546, (2010).
- Schwarz, G. Estimating the dimension of a model. *Annals of Statistics*, **7**(2),461-464, (1978).

# ***Capítulo 5***

**Algoritmos de Estimación de  
Distribuciones Aplicados al Diseño  
de Redes de Sensores**

## 5.1 Introducción

En este capítulo se presenta una nueva metodología para la resolución del Problema de Diseño de Redes de Sensores (PDRS) de costo mínimo sujeto a restricciones de estimabilidad y precisión sobre un conjunto de variables claves. La estrategia se basa en un Algoritmo de Estimación de Distribuciones (AEDs) dotado de búsqueda local.

El diseño de Redes de Sensores (RS) Generales involucra algunos aspectos importantes que condicionan necesariamente el algoritmo a utilizar para su resolución; entre éstos cabe mencionar:

- a) Necesidad de explorar un espacio de búsqueda de gran tamaño
- b) Posibilidad de existencia de múltiples subóptimos
- c) Necesidad de paralelizar su resolución, con el objeto de reducir los tiempos de computación en problemas muy grandes
- d) Capacidad de proveer buenas soluciones alternativas

Teniendo en cuenta estas premisas es que se propone desarrollar un nuevo AED que emplea estadísticos de orden uno como estrategia maestra, subordinada a la cual se aplica un mecanismo de búsqueda local basado en la técnica de Oscilación Estratégica (OS).

El análisis del comportamiento del algoritmo propuesto se basa en tres medidas de desempeño: la capacidad de encontrar soluciones de alta calidad, el esfuerzo computacional, y la posibilidad de escalamiento.



## 5.2 Algoritmo de Estimación de Distribuciones con Aprendizaje Incremental

Los Algoritmos Genéticos (AGs) tienen en cuenta las relaciones entre las variables de manera implícita, mientras que los AEDs las consideran explícitamente a través del modelo probabilístico asociado a los individuos de cada generación. Tal como se describió en el capítulo anterior, la estimación del modelo es una cuestión central de los AEDs, que ha motivado clasificarlos en tres categorías: univariados, bivariados y multivariados, según su complejidad. Para modelar la interdependencia de las variables, la segunda y tercera clases de los AEDs requieren procedimientos de aprendizaje complejos, los cuales en sí mismos resultan problemas de optimización, que insumen significativos recursos computacionales. Por esta razón, se considera conveniente volver la mirada hacia los AEDs más simples, aquellos que sólo utilizan estadísticos de orden uno, y realizar un estudio sobre cómo mejorar su desempeño para la resolución del problema de optimización bajo estudio.

Uno de los AEDs más difundidos, entre aquellos que suponen la independencia de la relación entre las variables, es el denominado *Population Based Incremental Learning*, PBIL, que constituye el núcleo de la estrategia de solución propuesta. Este procedimiento, presentado por Baluja (1994), fue originariamente ideado para obtener el óptimo de una función definida en un espacio binario  $I$ -dimensional, y más tarde fue mejorado por el mismo autor (Baluja y Caruana, 1995). Por su parte, Pang y col. (2006) y Wan y Qiu (2008), entre otros, han reportado buenos resultados de la aplicación del procedimiento en la resolución de problemas complejos.

En la Figura 5.1 se muestra el pseudocódigo del algoritmo PBIL. Como puede observarse, en primer lugar se procede a inicializar el vector de probabilidades  $\mathbf{p}^0$ , para luego ejecutar un procedimiento iterativo hasta que se satisfaga algún criterio de

detención previamente establecido. En la  $j$ -ésima iteración, se genera, por simulación, una población de  $M$  individuos teniendo en cuenta las probabilidades contenidas en  $\mathbf{p}^{j-1}$ . Cada individuo generado se evalúa utilizando la función de evaluación o función de *fitness*,  $F$ , y se determina cuál de ellos es la mejor solución,  $\mathbf{b}^j$ , en función de sus valores de  $F$ . A continuación se actualiza el vector de probabilidades, componente a componente utilizando para ello  $\mathbf{b}^j$  y la tasa de aprendizaje  $LR$  de la siguiente manera:

$$p_i^j = p_i^{j-1}(1 - LR) + b_i^j LR \quad (i = 1:I) \quad (5.1)$$

donde  $p_i^j$  es la componente  $i$ -ésima del vector de probabilidad actual y  $b_i^j$  representa el  $i$ -ésimo elemento de la mejor solución. Una vez producida la actualización del vector de probabilidades se lleva a cabo un procedimiento de mutación sobre  $\mathbf{p}^j$ , componente a componente, utilizando como parámetros la probabilidad de mutación  $PMUTA$  y el monto de la mutación  $MS$ . Para cada uno de los elementos en  $\mathbf{p}^j$  se genera un número aleatorio en el rango  $[0, 1]$  y, si éste es menor que  $PMUTA$  se aplica la siguiente expresión:

$$p_i^j = p_i^j (1 - MS) + rand(0,1)MS \quad (i = 1:I) \quad (5.2)$$

```

Inicializar el vector de probabilidades p
while Criterio de parada = FALSO
    Generar  $M$  individuos por simulación empleando p
    Calcular la función de evaluación  $F$  para cada uno de ellos
    Seleccionar la mejor solución b
    Actualizar p usando b y la tasa de aprendizaje  $LR$ 
    Mutar p un monto  $MS$  acorde con la probabilidad  $PMUTA$ 
end

```

Figura 5.1. Pseudocódigo de PBIL

En esta tesis se propone realizar el desarrollo e implementación del algoritmo, denominado pPBIL, el cual permite que *NPBIL* instancias puedan ejecutarse independientemente y que resulta intrínsecamente paralelizable. Una vez que las *NPBIL* subpoblaciones han evolucionado durante una iteración, sus vectores  $\mathbf{p}$  se relacionan, para intercambiar información entre ellos antes del muestreo, empleando mecanismos de combinación. Para cada par de vectores  $\mathbf{p}$ , la operación de combinación consiste en un cruzamiento uniforme aplicado con una probabilidad *Pinteracción*.

Un desempeño más eficiente del algoritmo pPBIL puede obtenerse implementando procedimientos de mejora basados esencialmente en mecanismos de búsqueda local. En la Figura 5.2 se presenta un esquema general del algoritmo híbrido.

```

Inicializar NPBIL vectores de probabilidad  $\mathbf{p}_k$  ( $k = 1, \dots, NPBIL$ )
while (criterio de parada = .FALSO.)
  for  $k = 1, \dots, NPBIL$ 
    Generar  $M$  individuos por simulación de acuerdo a  $\mathbf{p}_k$ 
    Calcular la función de evaluación  $F$  para cada miembro de la población
    Aplicar el Método de Mejora de soluciones a cada miembro de la población
    Seleccionar la mejor solución  $\mathbf{b}_k$ 
    Actualizar  $\mathbf{p}_k$  usando  $\mathbf{b}_k$  y la tasa de aprendizaje  $LR$ 
    Mutar  $\mathbf{p}_k$  usando una probabilidad de mutación  $PMUTA$  y un monto de
    Mutación  $MS$ 
  end
  OffSpring =  $\emptyset$ 
  for  $k = 1, \dots, NPBIL$  step 2
    Seleccionar 2 individuos (padres) entre todos los vectores  $\mathbf{p}$ 
    if random < Pinteracción
      Usar cruzamiento uniforme para generar dos descendientes
      Agregar descendientes al conjunto OffSpring
    end
  end
end

```

Figura 5.2. Pseudocódigo de pPBIL usando un procedimiento de mejora

```
        else
            Agregar padres a OffSpring
        end
    end
end
for  $k=1, \dots, NPBIL$ 
     $p_k = \text{OffSpring}(k)$ 
end
end
```

Figura 5.2. Continuación: Pseudocódigo de pPBIL usando un procedimiento de mejora

Dentro de los procedimientos de mejora podrían utilizarse los ya presentados en el Capítulo 3, los cuales han mostrado un desempeño aceptable para problemas de tamaño pequeño o medio. Sin embargo, la idea central aquí es incorporar un método de búsqueda local más eficiente, tal como OS. Esta es una metodología de búsqueda por entornos inspirada en la Búsqueda Tabú (BT) pero que presenta características particulares, las cuales se presentan a continuación.

### 5.3 Oscilación Estratégica

Búsqueda Tabú es un enfoque metaheurístico utilizado para resolver problemas de optimización que, en líneas generales, hace uso de un procedimiento de búsqueda local con memoria para explorar el espacio de soluciones. De esta manera se evita o previene el estancamiento en óptimos locales y la visita cíclica a las mismas soluciones.

Los aspectos básicos del procedimiento BT ya fueron explicados en el Capítulo 3. Allí se mencionó que la incorporación de otras técnicas al algoritmo de BT puede mejorar sus capacidades de intensificación y diversificación de la búsqueda. Con este fin se recurre a la metodología OS en esta tesis. Ésta consiste en una secuencia de fases destructivas y constructivas. Dada una solución inicial factible, en la primera fase la

búsqueda se guía estratégicamente de manera tal que cruza el límite de factibilidad y continúa dentro de la región no factible hasta cierta profundidad. La fase destructiva finaliza cuando una condición preestablecida se satisface (Condición 1). En esa situación, las reglas que gobiernan los movimientos cambian y la búsqueda procede hacia la región factible (fase constructiva) hasta que se verifica otra condición (Condición 2). En las fases destructiva y constructiva se inspeccionan vecindades diferentes de la solución actual representadas por  $\mathcal{N}_1(\mathbf{q})$  y  $\mathcal{N}_2(\mathbf{q})$ , respectivamente. En este sentido OS puede clasificarse como una búsqueda por entornos variables.

El procedimiento de cruzar repetidamente el límite de factibilidad desde diferentes regiones es lo que origina el comportamiento oscilatorio del método. Por otro lado, la aplicación de los mecanismos estándares de la BT, tales como las memorias de corto y largo plazo, evitan la réplica de las mismas trayectorias de manera cíclica. En la Figura 5.3 se muestra un pseudocódigo general para OS.

```
Entrada: solución inicial  $\mathbf{q}_0$ 
if  $\mathbf{q}_0$  es factible
    Asignar Fase=1 /* Fase destructiva */
else
    Asignar Fase=0 /* Fase constructiva */
end
Asignar  $\mathbf{q}^*=\mathbf{q}=\mathbf{q}_0$ 
for  $i=1$  to maxiter
    if Fase = 1
        Generar vecindad  $\mathcal{N}_1(\mathbf{q})$  y evaluar sus integrantes
        Elegir el mejor vecino  $\mathbf{q}'$ 
        if Condicion1=verdadero
            Fase = 0
        end
    end
```

Figura 5.3. Pseudocódigo para el algoritmo OS

```
else
    Generar vecindad  $\mathcal{N}_2(\mathbf{q})$  y evaluar
    Elegir el mejor vecino  $\mathbf{q}'$ 
    if Condicion2=verdadero
        Fase =1
    end
end
 $\mathbf{q}=\mathbf{q}'$ 
Actualizar listas tabú de memoria reciente y de frecuencia
if  $\mathbf{q} \prec \mathbf{q}^*$ 
     $\mathbf{q}^*=\mathbf{q}$ 
end
end
return  $\mathbf{q}^*$ 
```

Figura 5.3. Continuación: Pseudocódigo para el algoritmo OS

## 5.4 Nuevo Algoritmo Híbrido de Diseño de Redes de Sensores

A continuación se describen los aspectos fundamentales de la metaheurística pPBIL-OS desarrollada para resolver el PDRS formulado en el Capítulo 3. Dicha metaheurística combina una estrategia de búsqueda por entornos inspirada en BT con el algoritmo PBIL, con el fin de aprovechar sus respectivas fortalezas. La metaheurística pPBIL-OS (Carnero y col., 2013) se describe a continuación.

### 5.4.1 Función de Evaluación

La calidad de una RS se mide a través de la  $F$ . Ésta debe ser capaz de considerar la generación de soluciones no factibles tanto durante la ejecución de la técnica OS como del AED. Se propone utilizar la misma  $F$  descrita en el Capítulo 3 para abordar el PDRS empleando la BT, ya que satisface estos requerimientos. Así la  $F$  del procedimiento pPBIL-OS es:

$$F = \begin{cases} \sum_{i=1}^I c_i q_i & \text{si } \mathbf{q} \text{ es factible} \\ C(\mathbf{q}) & \text{si } \mathbf{q} \text{ es no factible} \end{cases} \quad (5.3)$$

La  $\sum_{i=1}^I c_i$  es el límite superior de  $F$ ; representa el costo de la red de sensores cuando todas las variables involucradas son medidas, y se denomina Límite de Factibilidad,  $FB$ . Por otra parte, la función  $C(\mathbf{q})$ , definida mediante la ecuación (3.14), tiene en cuenta que una solución es no factible cuando no se satisfacen las restricciones de observabilidad sobre alguna de las variables claves, o cuando todas éstas son estimables pero no se garantiza la precisión impuesta.

#### 5.4.2 Inicialización del Vector de Probabilidades

Cada solución potencial al PDRS,  $\mathbf{q} = \{q_1, q_2, \dots, q_I\}$ , se considera una realización de un vector aleatorio  $\mathbf{Q} = \{Q_1, Q_2, \dots, Q_I\}$  de dimensión  $I$ , donde  $Q_i$  es una variable aleatoria discreta que asume sólo dos valores posibles: 1(uno) si la  $i$ -ésima variable se mide o 0 (cero) en caso contrario. Si se considera que  $p_i$  es la función de distribución marginal correspondiente a  $Q_i$ , entonces  $\Pr(Q_i = q_i) = p_i(q_i)$  y, el modelo probabilístico que adopta el algoritmo PBIL puede factorizarse como el producto de las distribuciones marginales de cada variable involucrada, ya que éstas se consideran independientes. La distribución conjunta queda expresada entonces como:

$$f_{\mathbf{Q}}(q_1, q_2, \dots, q_I) = \prod_{i=1}^I p_i(q_i) \quad (5.5)$$

En otras palabras, el algoritmo sólo necesita realizar el aprendizaje paramétrico ya que la estructura del modelo está fija, y no se tienen en cuenta las interrelaciones entre las variables aleatorias intervinientes.

En algunos trabajos sobre el algoritmo PBIL, los elementos del vector de probabilidades  $\mathbf{p}$  se inicializan en 0.5, esto es, se considera que al comienzo no hay sesgo hacia soluciones particulares sino que todas son generadas al azar. En cambio, en esta tesis se propone incorporar conocimiento específico del problema para generar la población inicial. Para cada subpoblación del algoritmo pPBIL, se forma un conjunto  $\mathcal{S}^0$  de soluciones que satisfacen la estimabilidad de las variables claves y que poseen, además, cierto grado de diversidad estructural. El vector inicial de probabilidades,  $\mathbf{p}^0$  se estima entonces a partir de  $\mathcal{S}^0$ . El procedimiento consta de las siguientes etapas:

- 1) Generación de subpoblaciones previas  $\mathcal{S}$  que satisfagan las restricciones de estimabilidad de las variables claves. La metodología empleada en esta etapa es la desarrollada en el inciso 3.3.1.1.
- 2) Selección de las soluciones a incluir en  $\mathcal{S}^0$ . En principio se incorporan las  $M/2$  soluciones contenidas en  $\mathcal{S}$  con mejor valor de la  $F$ . Los restantes  $M/2$  miembros de  $\mathcal{S}^0$  se eligen calculando la distancia de Hamming entre las soluciones candidatas y cada una de las soluciones ya existentes en  $\mathcal{S}^0$ , y seleccionando la que resulta más distante.
- 3) Estimación del vector de probabilidades  $\mathbf{p}$ . La variable aleatoria  $Q_i$  tiene una distribución de Bernoulli con valor esperado igual a  $p_i$ . El estimador de máxima verosimilitud para este parámetro poblacional es la media muestral. Ésta se utiliza para calcular el valor de cada elemento del vector de probabilidades inicial. La componente  $p_i^0$  se determina a partir de  $\mathcal{S}^0$  como sigue:

$$p_i = \frac{1}{M} \sum_I^M q_i \quad (i=1:D) \quad (5.6)$$



A continuación se muestra un ejemplo de la metodología propuesta.

**Ejemplo 5.1:** se considera un proceso cuya operación se representa mediante un conjunto de ecuaciones de balance de masa total. Dicho conjunto se formula en términos de la siguiente matriz de incidencia

$$\mathbf{D} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & -1 \end{bmatrix}$$

Además se supone que se definen restricciones de estimabilidad y precisión para los conjuntos de caudales máxicos  $\mathcal{S}_e = \{z_1, z_2, z_6\}$  y  $\mathcal{S}_\sigma = \{z_2, z_6\}$ , respectivamente.

La Figura 5.4 muestra un conjunto  $\mathcal{AS}$  de potenciales soluciones. Para satisfacer la restricción  $E \geq 1$ , una variable debe ser medida o, no medida pero observable. Consecuentemente, cada miembro de  $\mathcal{AS}$  representa un conjunto de sensores que permite la estimación de todas las variables requeridas usando las mediciones directas o a través de los balances de masa. En la misma figura se muestra la población inicial  $\mathcal{D}^0$  para un tamaño  $M=6$ .

La estimación del vector  $\mathbf{p}^0$  en función de los miembros de  $\mathcal{D}^0$  se representa en la Figura 5.5.

# ind	$Q_1$	$Q_2$	$Q_3$	$Q_4$	$Q_5$	$Q_6$	$Q_7$	$Q_8$	$F$
1	1	1	1	1	1	1	1	1	13000
2	1	0	0	0	0	1	0	0	16500
3	1	1	1	0	0	1	0	0	7500
4	0	1	0	0	0	0	0	1	5500
5	1	1	0	0	0	0	0	1	17000
6	1	1	0	1	0	1	0	0	9000
7	1	1	0	0	0	1	0	0	18000
8	1	0	1	0	1	1	1	0	8500
9	1	1	1	0	0	1	0	0	7500
10	1	1	1	1	0	0	1	0	20500
11	1	0	0	1	1	1	0	0	6500

$\mathcal{P}$

# ind	$Q_1$	$Q_2$	$Q_3$	$Q_4$	$Q_5$	$Q_6$	$Q_7$	$Q_8$	$F$
4	0	1	0	0	0	0	0	1	5500
11	1	0	0	1	1	1	0	0	6500
3	1	1	1	0	0	1	0	0	7500
8	1	0	1	0	1	1	1	0	8500
1	1	1	1	1	1	1	1	1	13000
10	1	1	1	1	1	0	0	1	20500

$\mathcal{P}^0$

Figura 5.4. Construcción de la población inicial  $\mathcal{P}^0$  a partir de un conjunto previo  $\mathcal{P}$

# ind	$Q_1$	$Q_2$	$Q_3$	$Q_4$	$Q_5$	$Q_6$	$Q_7$	$Q_8$	$F$
4	0	1	0	0	0	0	0	1	5500
11	1	0	0	1	1	1	0	0	6500
3	1	1	1	0	0	1	0	0	7500
8	1	0	1	0	1	1	1	0	8500
1	1	1	1	1	1	1	1	1	13000
10	1	1	1	1	0	0	1	0	20500

$\mathcal{P}^0$

$$p_i = \frac{1}{M} \sum_{i=1}^M q_i \quad (i=1:J)$$

0.83	0.67	0.67	0.50	0.50	0.83	0.50	0.33
------	------	------	------	------	------	------	------

$\mathbf{p}^0$

Figura 5.5. Estimación del vector de probabilidades inicial  $\mathbf{p}^0$

### 5.4.3 Muestreo

En cada iteración del algoritmo PBIL se genera una nueva población muestreando el vector de probabilidades  $\mathbf{p}$  correspondiente a la iteración anterior. Con este propósito, se realiza una simulación de Monte Carlo de acuerdo al método de la transformación inversa, el cual utiliza un generador de números aleatorios uniformemente distribuidos,  $U$ , para simular una muestra aleatoria de la variable de interés.

Si se considera que la variable aleatoria  $Q_i$  asume dos valores, 0 o 1, con la siguiente distribución de probabilidades:

$$\begin{aligned} \Pr(Q_i = 1) &= p_i^{j-1} \\ \Pr(Q_i = 0) &= 1 - p_i^{j-1} \end{aligned} \tag{5.7}$$

entonces la función de distribución acumulada,  $H$ , tiene la siguiente expresión:

$$H(q_i) = \begin{cases} 0 & \text{si } q_i < 0 \\ 1 - p_i^{j-1} & \text{si } 0 \leq q_i < 1 \\ 1 & \text{si } 1 \leq q_i \end{cases} \quad (5.8)$$

y una muestra aleatoria de la variable  $Q_i$  puede generarse de acuerdo al siguiente esquema:

$$Q_i = \begin{cases} 0 & \text{si } U \leq 1 - p_i^{j-1} \\ 1 & \text{si } 1 - p_i^{j-1} < U \leq 1 \end{cases} \quad (5.9)$$

La población  $\mathcal{S}^j$  se construye simulando  $(M-1)$  muestras a partir del vector  $\mathbf{p}^{j-1}$  junto con el mejor individuo de la población  $(j-1)$ . Este mecanismo de elitismo se aplica a los fines de evitar la pérdida de buenas soluciones durante el proceso evolutivo. En la Figura 5.6 se puede observar la generación de la población  $\mathcal{S}^1$  mediante el muestreo del vector de probabilidades inicial  $\mathbf{p}^0$  correspondiente al Ejemplo 5.1.

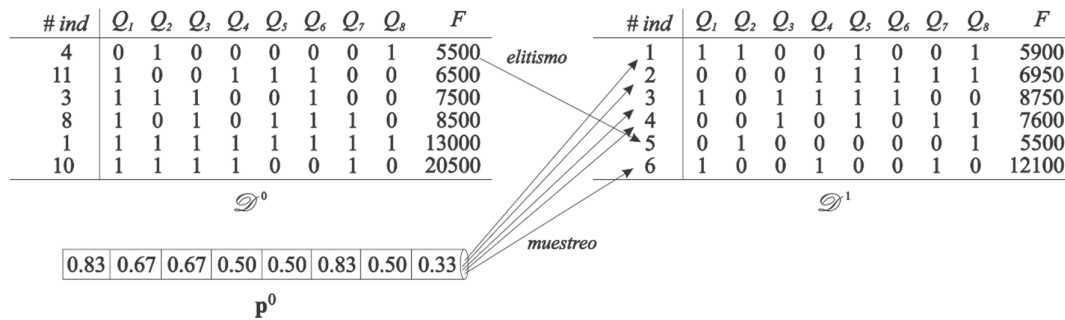


Figura 5.6. Generación de  $\mathcal{S}^1$  a partir de  $\mathbf{p}^0$

### 5.4.4 Búsqueda Local

Los beneficios asociados a la implementación de un procedimiento de búsqueda local dependen fuertemente de la calidad o estructura particular de la solución inicial y del costo computacional asociado a la técnica propuesta. Por lo tanto, la selección de cuáles y cuántas de las soluciones que forman la población del algoritmo PBIL son candidatas a punto inicial de la búsqueda local y la frecuencia con que este mecanismo se utiliza dentro del lazo evolutivo, son cuestiones importantes a ser resueltas.

Respecto de la selección de los individuos que serán sometidos a la búsqueda local (soluciones candidatas), la idea central es inspeccionar mediante el algoritmo de OS diferentes regiones dentro de cada subpoblación. Con este objetivo se ejecutan las siguientes etapas para cada una de ellas:

- 1) Identificación de la mejor solución corriente
- 2) Agrupamiento de los elementos de la subpoblación en  $nc$  grupos, *clusters*, usando una métrica de Hamming como medida de similaridad estructural y un algoritmo de  $K$ -medias para el procedimiento de agrupamiento.
- 3) Selección de la mejor solución corriente como una solución candidata
- 4) Selección aleatoria de otras soluciones candidatas, una por cada grupo, excepto para aquel grupo donde se encuentra la mejor solución ya seleccionada.

Dada una solución candidata  $\mathbf{q}$ , se construyen vecindades del tipo  $\mathcal{N}_1(\mathbf{q})$  o  $\mathcal{N}_2(\mathbf{q})$  de acuerdo a si el comportamiento oscilatorio está asociado a una etapa destructiva o a una constructiva, respectivamente. En el primer caso,  $\mathcal{N}_1(\mathbf{q})$  se define como el conjunto de soluciones obtenidas eliminando a partir de  $\mathbf{q}$  siempre una medición y cuando este movimiento no sea tabú. De manera similar a la heurística BTC, una Lista Tabú basada en lo Reciente almacena en un vector  $\mathbf{t}$ , de dimensión  $I$ , el número de iteraciones durante las cuales un movimiento está prohibido porque ha sido utilizado para obtener una solución reciente. La memoria de largo plazo del proceso está representada por el vector  $\mathbf{h}$ . La  $i$ -ésima componente de  $\mathbf{h}$  almacena el número de movimientos de la variable  $i$  usados para generar la próxima solución durante  $ph$  iteraciones. El valor de la  $F$  correspondiente al  $i$ -ésimo movimiento permitido se penaliza en proporción a  $h_i$ . Luego de  $ph$  iteraciones el vector  $\mathbf{h}$  se reinicializa.

Si la mejor solución vecina pertenece a un área tabú pero tiene un valor de  $F$  mejor que el de la mejor solución  $\mathbf{q}^*$ , entonces el estado tabú es revocado (criterio de aspiración). La Figura 5.7 muestra el pseudocódigo utilizado para generar la estructura de vecindad  $\mathcal{N}_1(\mathbf{q})$ .

*Entrada: solución  $\mathbf{q}$  y la mejor solución encontrada  $\mathbf{q}^*$*

$\mathcal{N}_1(\mathbf{q}) = \{\emptyset\}$

**for**  $i = 1$  to  $\text{length}(\mathbf{q})$

$\mathbf{q}' = \mathbf{q}$

**if**  $q'(i) = 1$

$q'(i) = 0$

**if** movimiento( $i$ ) es tabu

**if**  $\mathbf{q}' \prec \mathbf{q}^*$

                agregar  $\mathbf{q}'$  a  $\mathcal{N}_1(\mathbf{q})$

**end**

**else**

            agregar  $\mathbf{q}'$  to  $\mathcal{N}_1(\mathbf{q})$

**end**

**end**

**end**

**return**  $\mathcal{N}_1(\mathbf{q})$

Figura 5.7. Pseudocódigo para la generación de la vecindad  $\mathcal{N}_1(\mathbf{q})$

Durante la fase destructiva, la búsqueda cruza el  $FB$  y continúa en la región no factible hasta que se alcanza el valor  $L = \sum_{i=1}^I c_i + \max(\mathbf{c})$ .

Esto significa que la profundidad de la búsqueda en la región no factible es igual al costo del instrumento más caro disponible. En esta condición se inicia la fase constructiva incorporando mediciones a la solución corriente. En contraste con la fase anterior,  $\mathcal{N}_2(\mathbf{q})$  se define como el conjunto de soluciones obtenidas al agregar una medición a  $\mathbf{q}$

si este movimiento no es tabú. Cuando la búsqueda retorna a la región factible, la fase constructiva se detiene. En la Figura 5.8 se muestra el procedimiento de búsqueda local implementado para cada solución candidata.

```

Entrada: solution inicial  $\mathbf{q}_0$ 
 $L = \sum_{i=1}^I c_i + \max(\mathbf{c})$  y  $FB = \sum_{i=1}^I c_i$ 
if  $\mathbf{q}_0$  es factible
    Phase=1 /* fase destructiva */
else
    Phase=0 /* fase constructiva */
end
 $\mathbf{q}^* = \mathbf{q} = \mathbf{q}_0$ 
for  $i = 1$  to #maxiter do
    if Phase = 1
        Generar vecindad  $\mathcal{N}_1(\mathbf{q})$  y evaluar
        Obtener el mejor vecino  $\mathbf{q}'$ 
        if  $F(\mathbf{q}') > L$ 
            Phase = 0
        end
    else
        Generar vecindad  $\mathcal{N}_2(\mathbf{q})$  y evaluar
        Obtener el mejor vecino  $\mathbf{q}'$ 
        if  $F(\mathbf{q}') \leq FB$ 
            Phase=1
        end
    end
     $\mathbf{q} = \mathbf{q}'$ 
    Actualizar listas Tabú
    if  $\mathbf{q} \prec \mathbf{q}^*$ 
         $\mathbf{q}^* = \mathbf{q}$ 
    end
end
return  $\mathbf{q}^*$ 

```

Figura 5.8. Pseudocódigo del algoritmo OS propuesto para resolver el PDRS

Respecto del número de llamadas al algoritmo OS dentro del lazo evolutivo del algoritmo PBIL, es necesario analizar su impacto sobre la convergencia prematura a óptimos locales del AED. Una de las suposiciones subyacentes del método es que las buenas soluciones poseen bloques de estructuras similares. En el caso extremo de realizar una búsqueda local en el entorno de cada una de las soluciones de la población corriente,

toda la población resultante sería genotípicamente similar, la diversidad decaería abruptamente y con ello la posibilidad de salir de un valor subóptimo. Por otra parte si el número de veces que el AED utiliza OS es muy baja, no se estaría aprovechando la enorme capacidad de explotación que éste exhibe. En otras palabras se trata, como en la mayoría de las heurísticas conocidas, de establecer un balance adecuado entre la explotación del entorno de soluciones promisorias y la exploración inteligente del espacio de búsqueda.

Otro aspecto a tener en cuenta es la relación existente entre el costo computacional de la búsqueda por entorno, OS, y el tamaño del problema a resolver. La cardinalidad de la estructura del entorno de una solución,  $\mathcal{N}(\mathbf{q})$ , tanto en la fase constructiva como en la destructiva, es de orden  $I$  y cada elemento perteneciente a  $\mathcal{N}(\mathbf{q})$  debe ser evaluado un número de veces igual al número máximo de iteraciones permitidas,  $maxiter$ . Cada ejecución de OS insume, por lo tanto, un número de evaluaciones de la función objetivo del orden de  $maxiter * I$ . El tiempo de cómputo asociado a esta tarea puede ser excesivo en problemas grandes, de donde se deduce que la frecuencia con la que se admite que OS opere dentro del AED debe ser controlada.

En la metaheurística maestra propuesta, la frecuencia de aplicación de la heurística de búsqueda local subordinada se gobierna mediante el parámetro  $pso$ , que define la probabilidad que ocurra este evento en una subpoblación dada. Una vez que el algoritmo OS ha sido ejecutado, esta subpoblación se marca como inhibida de llamar a OS durante un número determinado,  $nso$ , de iteraciones.

**Ejemplo 5.2:** Se aborda el diseño de la red de caudalímetros de un proceso con 17 corrientes y 6 unidades. El conjunto de variables claves es  $\mathcal{S}_g = \{z_1, z_2, z_6\}$ . La Figura 5.9 muestra la aplicación, en una iteración dada, del procedimiento de búsqueda local a

una subpoblación con  $M=12$  y  $nc=2$ . El individuo #7 que pertenece al primer grupo, ( $ic=1$ ), es la mejor solución en la iteración actual, y por lo tanto se selecciona como solución candidata en este grupo. En el segundo conjunto, se lleva a cabo una selección al azar para determinar, entre todos sus integrantes, cuál será la solución candidata. Ésta corresponde a la solución #5. Ambos individuos, #7 y #5, son soluciones iniciales para el procedimiento OS que logra mejorarlos significativamente. En la Figura 5.10 se muestra el camino correspondiente a las transformaciones sufridas por las soluciones iniciales tanto en la fase constructiva como en la destructiva, en término de los valores de la  $F$  para  $maxiter=120$ . La Figura 5.11 muestra el camino seguido por las soluciones factibles y no factibles.

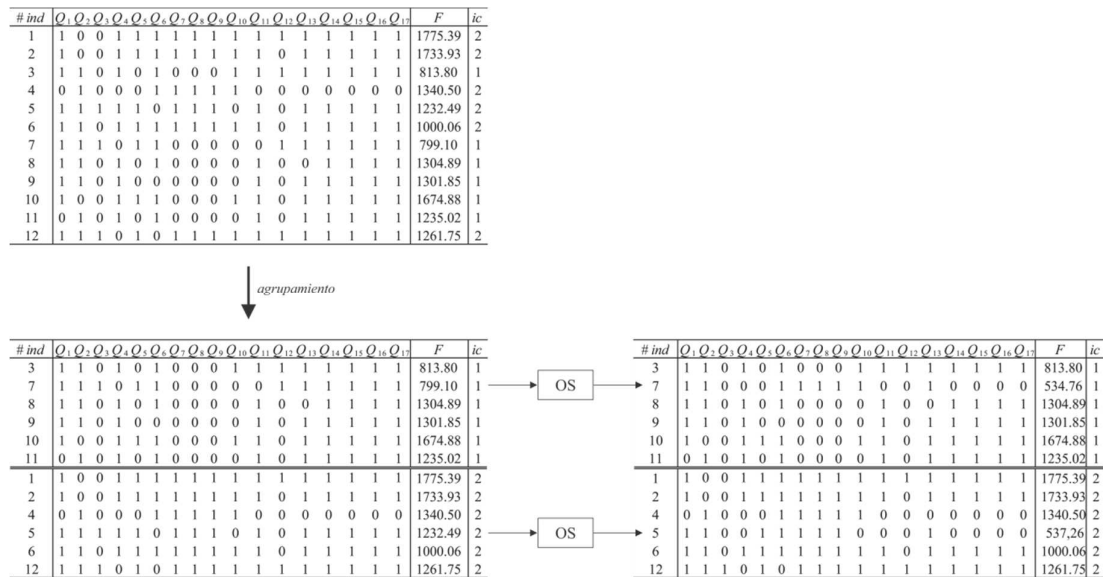


Figura 5.9. Agrupamiento y mejora utilizando el algoritmo OS



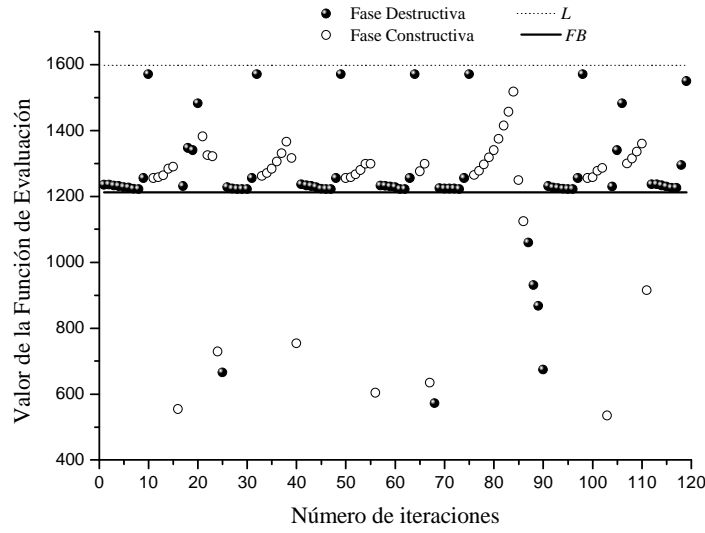


Figura 5.10. Recorrido de las soluciones en las fases constructivas y destructivas del algoritmo OS

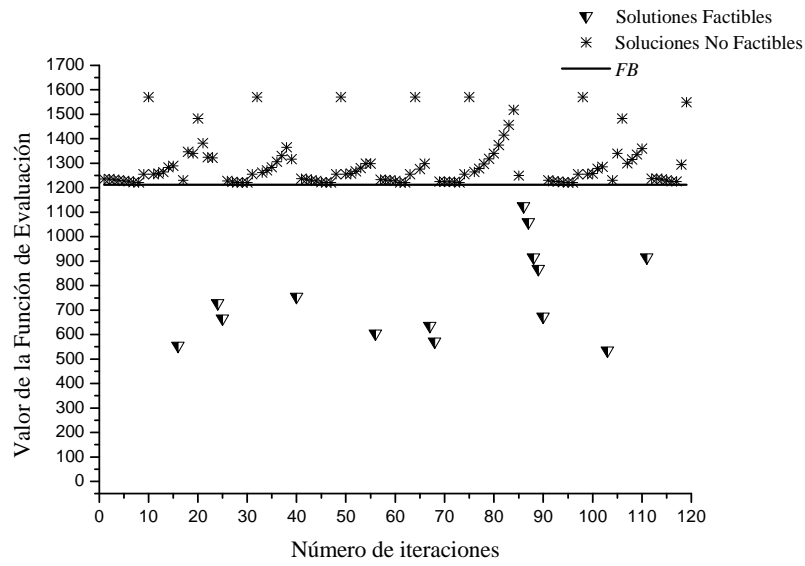


Figura 5.11. Recorrido de las soluciones factibles y no factibles en el algoritmo OS

### 5.4.5 Pseudocódigo

El pseudocódigo del algoritmo propuesto se presenta en la Figura 5.12. Cada subpoblación se caracteriza por las siguientes variables:  $\mathbf{p}$  es el vector de probabilidad, cada fila de la matriz  $\mathbf{S}$  almacena las soluciones potenciales del problema,  $\mathbf{fit}$  es el vector que contiene los valores de  $F$  para cada solución en  $\mathbf{S}$ ,  $\mathbf{b}$  es la mejor solución en  $\mathbf{S}$ ,  $\mathbf{I\_list}$  es un vector de dimensión  $NPBIL$  cuyos elementos guardan el número de iteraciones restantes antes que la aplicación de la heurística OS sea prohibida para una subpoblación dada. El algoritmo se detiene cuando se alcanza un número preestablecido,  $MaxGen$ , de iteraciones.

```

/* Inicializar PBILs */
I_list=0          /*Inicializar lista de inhibición */
for  $k=1$  to  $NPBIL$ 
    generar subpoblación previa  $\mathcal{S}$ 
     $\mathcal{S}^0(1..M/2)=Mejores\_Soluciones(\mathcal{S}, M/2)$ 
     $\mathcal{S}^0(M/2+1..M)=Soluciones\_más\ distantes(\mathcal{S}, M/2)$ 
     $\mathbf{S}_k=\mathcal{S}^0$ 
    for  $i=1$  to  $I$ 
         $\mathbf{p}_k(i) = \text{sum}(\mathbf{D}_k(m,i)) / M \quad m=1..M$ 
    end
end
for  $j=1$  to  $MaxGen$ 
    for  $k=1$  to  $NPBIL$ 
         $\mathbf{S}_k = \text{muestrear}(\mathbf{p}_k)$ 
        for  $m=1$  to  $M$ 
             $\mathbf{fit}_k = F(\text{fila}_m(\mathbf{S}_k))$ 
        end
         $\mathbf{b}_k = \text{mejor individuo en } \mathbf{S}_k$ 
        if  $PBIL_k$  no está inhibido y  $\text{rand} < pso$ 
            Dividir  $\mathbf{S}_k$  en  $nc$  clusters
    
```

Figura 5.12. Pseudocódigo para el algoritmo pPBIL-OS

```

        clbest=cluster donde está  $\mathbf{b}_k$ 
        Aplicar OS a  $\mathbf{b}_k$ 
        for  $w=1$  to  $nc$ 
            if  $w \neq clbest$ 
                Seleccionar al azar una solución  $\mathbf{q}_w$ 
                Aplicar OS a  $\mathbf{q}_w$ 
            end
        end
        actualizar  $\mathbf{S}_k$ 
        actualizar  $\mathbf{b}_k$ 
         $\mathbf{I\_list}(k) = nso$   /* inhibir OS en PBILk por nso iteraciones */
    end
    Actualizar vector de probabilidad  $\mathbf{p}_k$  con tasa de aprendizaje LR
    Mutar  $\mathbf{p}_k$  usando una probabilidad de mutación PMUTA y una cantidad
    de mutación MS
    Actualizar  $\mathbf{I\_list}$ 
end
if  $randnumber < Pinteracción$ 
    Modificar  $\mathbf{p}_k$  vectores mediante el operador de cruzamiento
end
end
sol_salida=mejor(b)
return(sol_salida)

```

Figura 5.12. Continuación: Pseudocódigo para el algoritmo pPBIL-OS

## 5.5 Ejemplos y Resultados

En este apartado se muestran los resultados obtenidos, utilizando el algoritmo propuesto, para la resolución de los casos Diseño 1 y Diseño 2 presentados en los capítulos previos

En las Tablas 5.1 y 5.2 se presentan los parámetros asignados al algoritmo pPBIL-OS, y los datos y la mejor solución encontrada para el Diseño 1.

Parámetro	Valor	Parámetro	Valor
<i>NPBIL</i>	4	<i>pso</i>	0.05
<i>M</i>	12	<i>nso</i>	25
<i>MaxGen</i>	200	<i>nc</i>	2
<i>LR</i>	0.1	<i>maxiter</i>	120
<i>PMUTA</i>	0.02	<i>pt</i>	$1.5\sqrt{n}$
<i>MS</i>	0.05	<i>ph</i>	$n/2$
<i>Pinteraccion</i>	0.25		

Tabla 5.1. Parámetros del algoritmo pPBIL-OS

Variables claves	$\sigma_i^*$	$\hat{\sigma}_i$	Variables Medidas	<i>F</i>
1	-----	$\hat{\sigma}_1=0.0215$	1 2 6 7 9 10 13 20 26 28	533.56
2	$\sigma_2^*=0.025$	$\hat{\sigma}_2=0.025$		
6	$\sigma_6^*=1.7851$	$\hat{\sigma}_6=1.6552$		

Tabla 5.2. Datos y Resultados para el Diseño 1 - pPBIL-OS

Se analizó además el comportamiento del algoritmo en términos de la repetitividad de la mejor solución obtenida. Dado que el método de solución propuesto tiene varios componentes aleatorios (población inicial, número de veces que se aplica la búsqueda local y el muestreo de las potenciales soluciones), para evaluar su desempeño debe estudiarse su capacidad de replicar, en diferentes corridas, los mejores valores encontrados. Con este objetivo se realizaron 100 experiencias diferentes, y los resultados del análisis estadístico de las mejores soluciones obtenidas se muestran en la Tabla 5.3.

Valor mínimo	Media	Desvío	CV(%)
533.56	533.56	0	0

Tabla 5.3. Estadísticos de la solución para el Diseño 1 - pPBIL-OS

En relación con el Diseño 2, en las Tablas 5.4 y 5.5 se presentan los datos y la mejor solución encontrada, y los resultados del análisis estadístico de las mejores soluciones obtenidas para 100 corridas del algoritmo, respectivamente.

VARIABLES CLAVES	$\sigma_i^*$	$\hat{\sigma}_i$	VARIABLES MEDIDAS	$F$
2	-----	$\hat{\sigma}_2=0.025$	1 2 5-10 13 19 20 23 26-28	894.86
10	$\sigma_{10}^*=1.0482$	$\hat{\sigma}_{10}=0.8861$		
28	$\sigma_{28}^*=1.4446$	$\hat{\sigma}_{28}=1.435$		

Tabla 5.4. Datos y Resultados para el Diseño 2 - pPBIL-OS

Valor mínimo	Media	Desvío	CV(%)
894.86	894.86	0	0

Tabla 5.5. Estadísticos de la solución para el Diseño 2 - pPBIL-OS

Puede observarse que el desempeño del algoritmo para los dos problemas de optimización es similar: el mejor valor se obtiene en el 100 % de los experimentos realizados.

Tal como se ha hecho notar en los capítulos anteriores, el Diseño 2 es el más más complejo, y esto motiva que los métodos de resolución basados sólo en búsqueda local o puramente evolutivos presenten CV del orden del 24% para este caso de estudio. Si el Diseño 2 se resuelve empleando BD, el CV disminuye a 17 % , mientras que para el algoritmo pPBIL-OS es cero. Es decir, el algoritmo converge a un mismo valor cada vez que se ejecuta, y además este valor coincide con la mejor solución encontrada.

## 5.6 Conclusiones

Dada las características complejas del PDRS planteado y la necesidad de resolver instancias del mismo que representen procesos de tamaño real, en este capítulo se ha propuesto, como técnica de resolución, una metaheurística que combina aspectos de un algoritmo evolutivo con un método de búsqueda por entornos.

El nuevo procedimiento desarrollado emplea un número predeterminado de subpoblaciones que evolucionan de manera independiente, pudiendo intercambiar información entre ellas con cierta probabilidad. En cada una de estas subpoblaciones actúan de manera controlada mecanismos de búsqueda local que aseguran una inspección más sesgada hacia zonas promisorias del espacio de soluciones.

El comportamiento de la técnica de resolución propuesta, analizado para problemas de tamaño medio (28 variables binarias en total) es ampliamente satisfactorio. A pesar de la naturaleza probabilística del algoritmo pPBIL-OS, el conjunto de las soluciones obtenidas en diferentes corridas del procedimiento muestra una dispersión igual a cero y una convergencia hacia buenas soluciones.

## 5.7 Nomenclatura

<b>b</b>	Mejor individuo
<b>C</b>	Función de penalización
<b>c</b>	Vector de costos de instrumentación.
<b>D</b>	Jacobiano de las ecuaciones del modelo linealizado
$\mathcal{P}$	Población de individuos

---

$E_{le}$	Grado de estimabilidad de la variable $le$
$F$	Función de evaluación
$FB$	Límite de factibilidad
<b>fit</b>	Vector de valores de la función de evaluación
$f_Q$	Distribución de probabilidad conjunta
$H$	Función de distribución acumulada
<b>h</b>	Lista tabú basada en la frecuencia
$I$	Número de variables de proceso
<b>I_list</b>	Lista de inhibición
$ic$	Índice del grupo
$J$	Número de ecuaciones del modelo del proceso
$L$	Límites para OS
$LR$	Tasa de aprendizaje
$M$	Número de individuos para cada instancia del algoritmo PBIL
$MaxGen$	Número máximo de iteraciones del algoritmo PBIL
$maxiter$	Número máximo de iteraciones para OS
$MS$	Cantidad de mutación
$\mathcal{N}$	Vecindad de posibles soluciones

---

$nc$	Número de grupos
$NPBIL$	Número de instancias independientes del algoritmo PBIL
$nso$	Número de iteraciones permitidas entre llamadas a OS
$\mathbf{p}$	Vector de probabilidad
$ph$	Número de iteraciones antes que $\mathbf{h}$ se reinicie
$Pinteraccion$	Probabilidad de cruzamiento
$PMUTA$	Probabilidad de mutación
$\mathcal{P}$	Población previa
$Pr$	Probabilidad
$pso$	Probabilidad de llamada al procedimiento OS
$pt$	Período tabú
$pt$	Período tabú para la memoria de corto plazo
$\mathbf{Q}$	Vector aleatorio $I$ -dimensional
$\mathbf{q}$	Vector de variables binarias
$\mathbf{S}$	Matriz de soluciones
$\mathcal{L}_\theta$	Conjunto de variables de proceso sujetas a restricciones de estimabilidad
$\mathcal{L}_\sigma$	Conjunto de variables de proceso sujetas a restricciones de precisión



$U$  Variable aleatoria con distribución uniforme

$\mathbf{t}$  Lista tabú basada en lo reciente

$\mathbf{z}$  Vector de caudales máxicos

Letras Griegas

$\sigma_i$  Desvío estándar de la  $i$ -ésima medición

$\hat{\sigma}_p$  Desvío estándar de la estimación de la  $lp$ -ésima variable

$\sigma_p^*$  Desvío estándar superior de la estimación de la  $lp$ -ésima variable

## 5.8 Acrónimos

AE Algoritmo Evolutivo

AED Algoritmo de Estimación de Distribuciones

AG Algoritmo Genético

PBIL *Population Based Incremental Learning*

pPBIL Algoritmo PBIL paralelo

PDRS Problema de Diseño de Redes de Sensores

OS Oscilación Estratégica

## 5.9 Referencias

- Baluja S, Caruana R. Removing the genetics from the standard genetic algorithm. *Technical Report CMU-CS-95-141*, Carnegie Mellon University, (1995).
- Baluja S. Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning. *Technical Report CMU-CS-94-163*, Carnegie Mellon University, (1994).
- Carnero M, Hernández J, Sánchez M. A new metaheuristic based approach for the design of sensor networks. *Comp. Chem. Eng.*, **55**, 83-96, (2013).
- Pang H, Hu K, Hing Z. Adaptive PBIL algorithm and its application to solve scheduling problems. *IEEE Conference on Computer Aided Control System Design*, Munich, Germany, 784-789, (2006).
- Wan S, Qiu D. Vehicle routing optimization problem with time constraint using advanced PBIL algorithm. *IEEE International Conference on Service Operations and Logistics, and Informatics*, Beijing, China, **1**, 1394-98, (2008).

# ***Capítulo 6***

**Ejemplos de Aplicación**

## 6.1 Introducción

En este capítulo se presentan los resultados obtenidos utilizando diferentes estrategias evolutivas, basadas en Algoritmos de Estimación de Distribuciones (AEDs), para la resolución del Problema de Diseño de Redes de Sensores (PDRS) que se aborda en esta tesis, (Carnero y col., 2015). Se emplean tres casos de estudio, con el fin de evaluar el desempeño de las técnicas, que comprenden procesos con diferentes números de equipos y corrientes. A continuación se los describe en orden creciente de tamaño y grado de especificación respecto de las variables requeridas.

En el Caso 1 se considera nuevamente el sistema de vapor de una planta de producción de metanol, que consta de 11 equipos y 28 corrientes, descrito en el Capítulo 3. Se analizan en detalle dos diseños que involucran conjuntos de variables claves distintos a los ya evaluados. Los Casos 2 y 3 se refieren a procesos que comprenden 19 unidades y 52 corrientes, y 47 equipos y 82 corrientes, respectivamente.

Los ejemplos contemplan la ubicación de sensores sobre cualquier corriente, suposición que aumenta el número de combinaciones a evaluar por el algoritmo de resolución. Los tamaños de los espacios de búsqueda son  $2^{28}$ ,  $2^{52}$  y  $2^{82}$ , es decir, los números de combinaciones posibles son del orden de  $10^6$ ,  $10^{15}$  y  $10^{24}$  para los Casos 1 a 3, respectivamente.

Se analizan los resultados alcanzados al utilizar el algoritmo pPBIL-OS, propuesto en el capítulo anterior, para resolver los casos de estudio, y se los compara con los obtenidos por otros AEDs. Entre éstos se eligen los algoritmos multivariados AffEDA, *Affinity Propagation Estimation Distribution Algorithm* (Santana y col., 2010) y EBNA, *Estimation of Bayesian Network Algorithm* (Etxeberria y Larrañaga, 1999). El primero emplea la descomposición de la distribución conjunta de probabilidades de un vector

aleatorio en un producto de múltiples distribuciones de probabilidades y la segunda realiza un aprendizaje estructural basado en Redes Bayesianas (RB).

Con el objetivo de evaluar el desempeño de las diferentes heurísticas, se diseñaron 100 experiencias independientes de cada algoritmo para luego analizar los siguientes estadísticos:

- a) Valor mínimo encontrado (Mínimo): el mejor valor encontrado en el proceso de optimización sirve obviamente como valor de referencia, ya que en el conjunto de problemas analizados no se conoce el valor del óptimo global ni límites inferiores para la función objetivo.
- b) Media: el mínimo de la muestra de 100 experiencias por sí sólo no brinda toda la información deseada. La media permite analizar la tendencia central de los valores obtenidos en la minimización.
- c) Mediana y percentiles: el algoritmo tiene mejor desempeño cuanto más cercanos se encuentren los valores de la mediana y del 99 percentil respecto del mínimo valor, ya que entonces es capaz de: replicar el mejor valor o bien hallar “buenas soluciones” en cada ejecución del mismo.
- d) Desvío estándar (Desvío) y coeficiente de variación (CV): aportan información complementaria a la previamente citada, y en el caso del CV, como una medida adimensional de dispersión relativa a la media, va a permitir comparar la variabilidad de los resultados obtenidos cuando éstos tienen valores medios de órdenes de magnitud diferentes.

Todos los ejemplos han sido ejecutados utilizando un Procesador Intel® Core (TM) i7 CPU 920 @ 2.67 GHz, 6GB RAM, usando MatLab Release 14. La implementación en paralelo del algoritmo fue simulada mediante corridas secuenciales de *NPBIL* instancias

del procedimiento pPBIL-OS. La comunicación entre cada procedimiento pPBIL-OS se realizó a través de la actualización de los vectores de probabilidad mediante un mecanismo de cruzamiento. Por lo tanto los tiempos de cómputo reportados a continuación revisten sólo un carácter ilustrativo y propósitos comparativos.

### 6.2 Caso 1

En la Tabla 6.1 se muestran los requerimientos considerados para los dos diseños de la red de caudalímetros máxicos, correspondientes al proceso de 11 equipos y 28 corrientes, denominados D3 y D4, respectivamente.

Caso 1	Especificaciones
	$E_{le} \geq 1$ para las corrientes 17 23
D3	$\sigma_4^* = 2.199$ $\sigma_8^* = 3.281$ $\sigma_{21}^* = 1.754$ $\sigma_{25}^* = 1.709$
	$E_{le} \geq 1$ para las corrientes 7 16 18 20
D4	$\sigma_4^* = 2.199$ $\sigma_5^* = 1.065$ $\sigma_8^* = 3.281$ $\sigma_{12}^* = 1.345$
	$\sigma_{27}^* = 1.415$ $\sigma_{28}^* = 1.445$

Tabla 6.1. Especificaciones de los diseños – Caso 1

Se realizaron conjuntos de 100 experiencias con tamaño de población  $M= 50$ , para los algoritmos EBNA y AffEDA, y con  $M=12$  y  $NPBIL=4$  (48 individuos en total) para pPBIL-OS, de manera tal de poder establecer comparaciones sobre tamaños de poblaciones similares.

Para D3 el mejor valor encontrado corresponde a un valor de la Función de Evaluación,  $F$ , igual a 752.26 e implica medir las variables 1, 4, 6, 7, 9, 10, 11, 14, 16-

24. La precisión alcanzada sobre las variables claves con este conjunto de sensores es  $\hat{\sigma}_4=2.1857$ ,  $\hat{\sigma}_8=2.5644$ ,  $\hat{\sigma}_{17}=0.8048$ ,  $\hat{\sigma}_{21}=1.5018$ ,  $\hat{\sigma}_{23}=0.0648$ ,  $\hat{\sigma}_{25}=1.4864$ . En la Tabla 6.2 se muestran los valores resultantes del proceso de optimización utilizando los diferentes algoritmos. Por su parte, en la Figura 6.1 se grafican los valores mínimos de  $F$  obtenidos en cada una de las 100 corridas en orden ascendente.

Metaheurística	Mínimo	Media	Mediana	99 Percentil	Desvío	CV
pPBIL-OS	752.26	752.26	752.26	752.26	0	0
AffEDA	752.26	763.75	754.36	850.76	19.13	0.025
EBNA	754.76	1511.3	1801.6	1832.5	343.25	0.227

Tabla 6.2. Resultados obtenidos para D3 - pPBIL-OS, AffEDA y EBNA

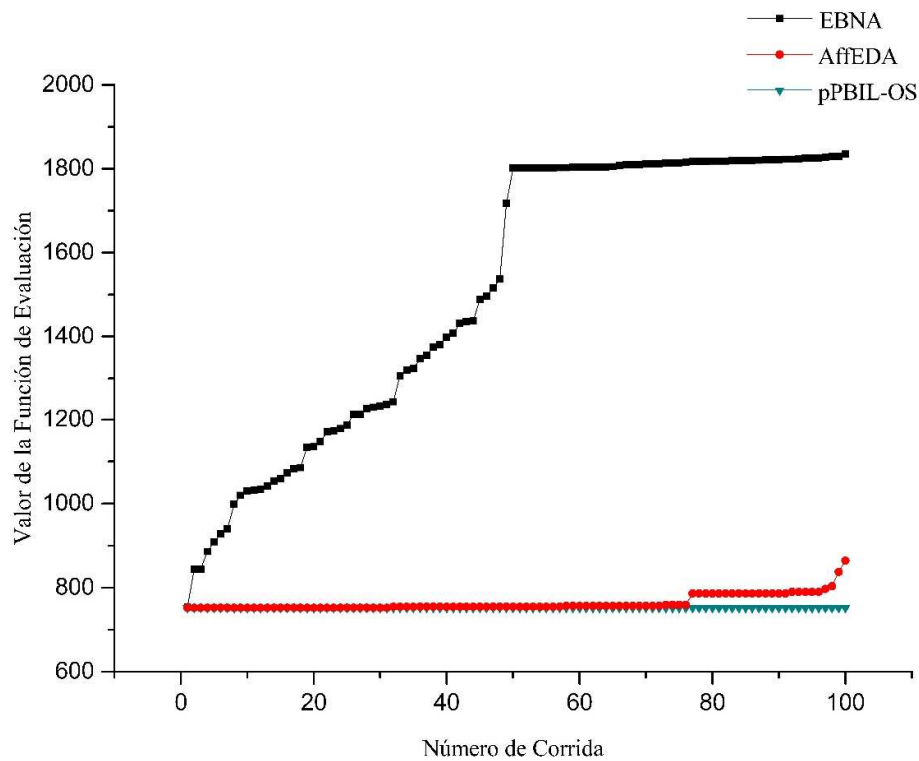


Figura 6.1. Mejores valores encontrados para D3 - pPBIL-OS, AffEDA y EBNA

El caso de estudio D4 corresponde a una instancia del problema que requiere un conjunto mayor de variables claves sujetas a restricciones tanto de observabilidad como de precisión. Constituye un diseño con un grado de especificación más alto que el D3. El mejor valor encontrado para la  $F$  es igual a 1178.06 e implica medir las variables 1, 2, 4, 5-7, 9-11, 13, 15-24, 26-28. La precisión alcanzada sobre las variables claves para esta red de sensores es  $\hat{\sigma}_4=2.0368$ ,  $\hat{\sigma}_5=0.8878$ ,  $\hat{\sigma}_7=0.0580$ ,  $\hat{\sigma}_8=1.4967$ ,  $\hat{\sigma}_{12}=0.9588$ ,  $\hat{\sigma}_{16}=0.5848$ ,  $\hat{\sigma}_{18}=0.3965$ ,  $\hat{\sigma}_{20}=0.2614$ ,  $\hat{\sigma}_{27}=1.2002$ ,  $\hat{\sigma}_{28}=1.4437$ . En la Tabla 6.3 se resumen los resultados obtenidos, mientras que en la Figura 6.2 se grafican los valores mínimos de  $F$  alcanzados en cada corrida previamente ordenados.

Metaheurística	Mínimo	Media	Mediana	99 Percentil	Desvío	CV
pPBIL-OS	1178.06	1178.06	1178.06	1178.06	0	0
AffEDA	1178.06	1271.4	1224.11	1798.85	141.36	0.11
EBNA	1288.86	1722.37	1805.88	1821.19	157.7	0.091

Tabla 6.3. Resultados obtenidos para D4 - pPBIL-OS, AffEDA y EBNA



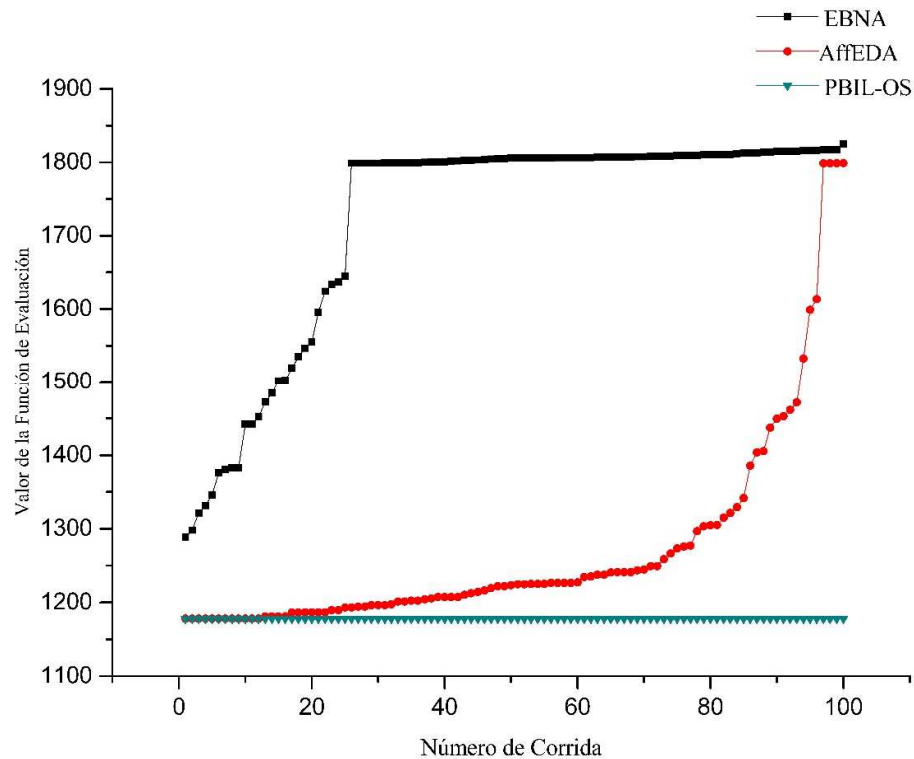


Figura 6.2. Mejores valores encontrados para D4 - pPBIL-OS, AffEDA y EBNA

En ambos tipos de diseños, el desempeño del algoritmo pPBIL-OS es claramente superior al que exhiben los otros dos AEDs probados, ya que alcanza el mejor valor encontrado en el 100 % de las ejecuciones. El algoritmo EBNA es el más pobre en cuanto a la calidad de las soluciones obtenidas, no logra encontrar el mínimo de referencia en el caso de estudio D4, y el resto de los estadísticos calculados presentan valores muy superiores comparados con los provistos por los otros procedimientos. Por último, AffEDA es eficiente para la resolución del diseño D3 y proporciona valores de los estadísticos comparables a los obtenidos empleando pPBIL-OS. Sin embargo, la eficiencia del método AffEDA decae para el diseño D4 ya que sólo puede proveer el mínimo valor encontrado (1178.06) en el 10 % de las corridas, siendo la media y el desvío estándar iguales a 1271.4 y 141.36, respectivamente.

En la Figura 6.3 y en la Tabla 6.4 se presentan los tiempos de ejecución asociados a cada una de las 100 corridas y sus estadísticos, respectivamente, para el diseño D3. El criterio de parada para cada uno de los procedimientos es el número de iteraciones que se fijó en 200.

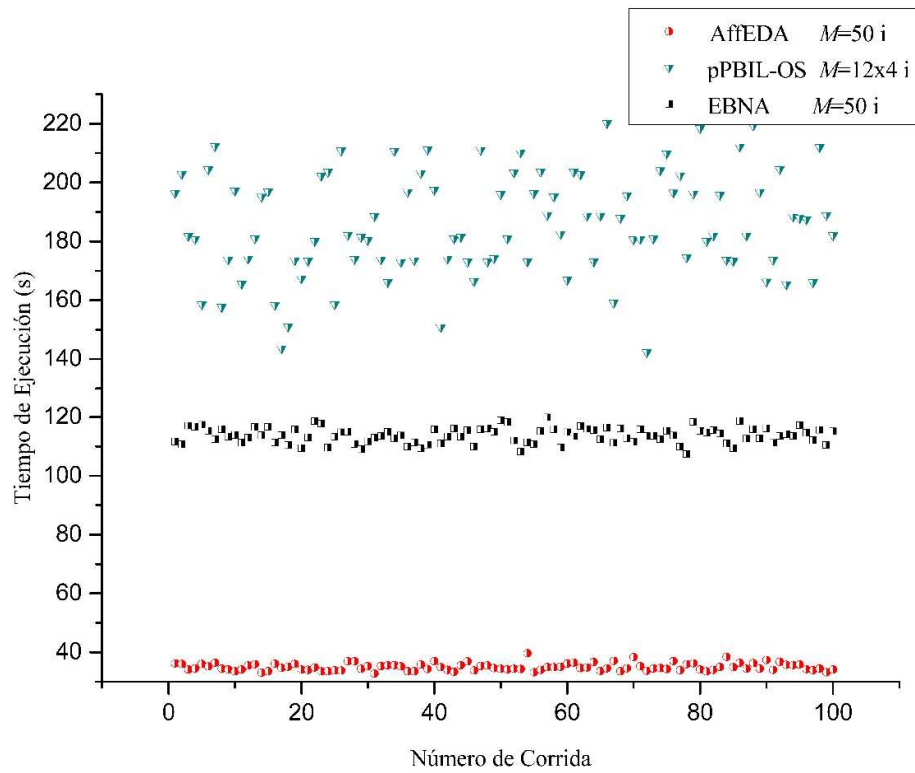


Figura 6.3. Tiempos de Ejecución para D3 - pPBIL-OS, AffEDA y EBNA

Metaheurística	Media	Desvío	Mínimo	Mediana	Máximo
pPBIL-OS	185.38	17.53	142.23	181.95	220.10
AffEDA	35.02	1.29	32.74	36.67	39.66
EBNA	113.72	2.67	107.36	113.67	119.87

Tabla 6.4. Estadísticos de los tiempos de ejecución – D3

Un análisis similar se presenta en la Figura 6.4 y Tabla 6.5 para el diseño D4.

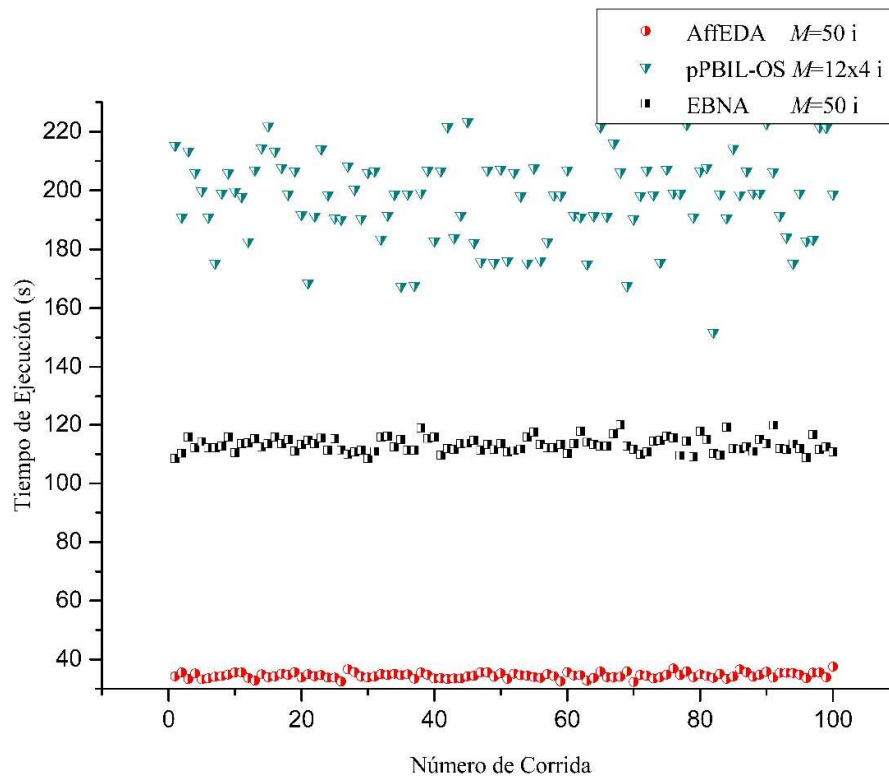


Figura 6.4. Tiempos de Ejecución para D4 - pPBIL-OS, AffEDA y EBNA

Metaheurística	Media	Desvío	Mínimo	Mediana	Máximo
pPBIL-OS	197	14.7	15.73	198.70	223.30
AffEDA	34,54	0.98	32.34	34.42	37.54
EBNA	113.15	2.50	108.50	112.67	119.93

Tabla 6.5. Estadísticos de los tiempos de ejecución – D4

Para ambos diseños el algoritmo pPBIL-OS es el que insume más tiempo de cómputo; le siguen en orden decreciente EBNA y AffEDA. Los tiempos de ejecución asociados a cada corrida están relacionados con el número de veces que se calcula la  $F$ . Para los métodos EBNA y AffEDA, la cantidad de evaluaciones de dicha función depende sólo del tamaño de la población. En cambio, el algoritmo pPBIL-OS tiene una componente aleatoria adicional que es el llamado al procedimiento Oscilaciones Estratégicas (OS), lo cual explica la variabilidad observada en sus tiempos de cómputo.

### 6.3 Caso 2

Este caso corresponde a la unidad de destilación de crudo, presentada en la Figura 6.5 (Gala y Bagajewicz, 2006; Nguyen y Bagajewicz, 2011). El proceso consta de 19 unidades y 52 corrientes. En la Tabla 6.6 se incluye los datos asignados a los caudales verdaderos de cada corriente,  $C^{m_i}$ , y los desvíos estándares ( $\sigma_i$ ) y costos ( $c_i$ ) de los caudalímetros másicos.

Corriente	$C^{m_i}$	$\sigma_i \cdot 10^3$	$c_i$
1	413349	8.2670	826.7001
2	419579	8.3916	839.1601
3	209313	4.1863	418.6281
4	210262	4.2052	420.5261
5	419579	8.3916	839.1601
6	460520	9.2104	921.0421
7	26510	0.5302	53.0221
8	230650	4.6130	461.3021
9	229870	4.5974	459.7421
10	26243	0.5249	52.4881
11	413650	8.2730	827.3021
12	413650	8.2730	827.3021
13	206932	4.1386	413.8661
14	206717	4.1343	413.4361
15	413650	8.2730	827.3021
16	27068	0.5414	54.1381
17	5124	0.1025	10.2501
18	21467	0.4293	42.9361
19	478	0.0096	0.9581
20	61562	1.2312	123.1261
21	60985	1.2197	121.9721

Tabla 6.6. Datos - Caso 2

22	61253	1.2251	122.5081
23	61490	1.2298	122.9821
24	61109	1.2222	122.2201
25	60796	1.2159	121.5941
26	62012	1.2402	124.0261
27	60413	1.2083	120.8281
28	103939	2.0788	207.8801
29	386580	7.77316	773.1621
30	57169	1.1434	114.3401
31	45829	0.9166	91.6601
32	4202	0.0840	8.4061
33	26133	0.5227	52.2681
34	73900	1.4780	147.8021
35	73704	1.4741	147.4101
36	50851	1.0170	101.7041
37	50715	1.0543	101.4321
38	45902	0.9180	91.8061
39	45878	0.9176	91.7581
40	45928	0.9186	91.8581
41	45851	0.9170	91.7041
42	185593	3.7119	371.1881
43	38557	0.7711	77.1161
44	18932	0.6786	37.8661
45	19846	0.3969	39.6941
46	23880	0.4776	47.7621
47	18196	0.3639	36.3941
48	18106	0.3621	36.2141
49	48081	0.9616	96.1641
50	15154	0.3031	30.3101
51	20268	0.4054	40.5381
52	12659	0.2532	25.3201

Tabla 6.6. Continuación: Datos - Caso 2

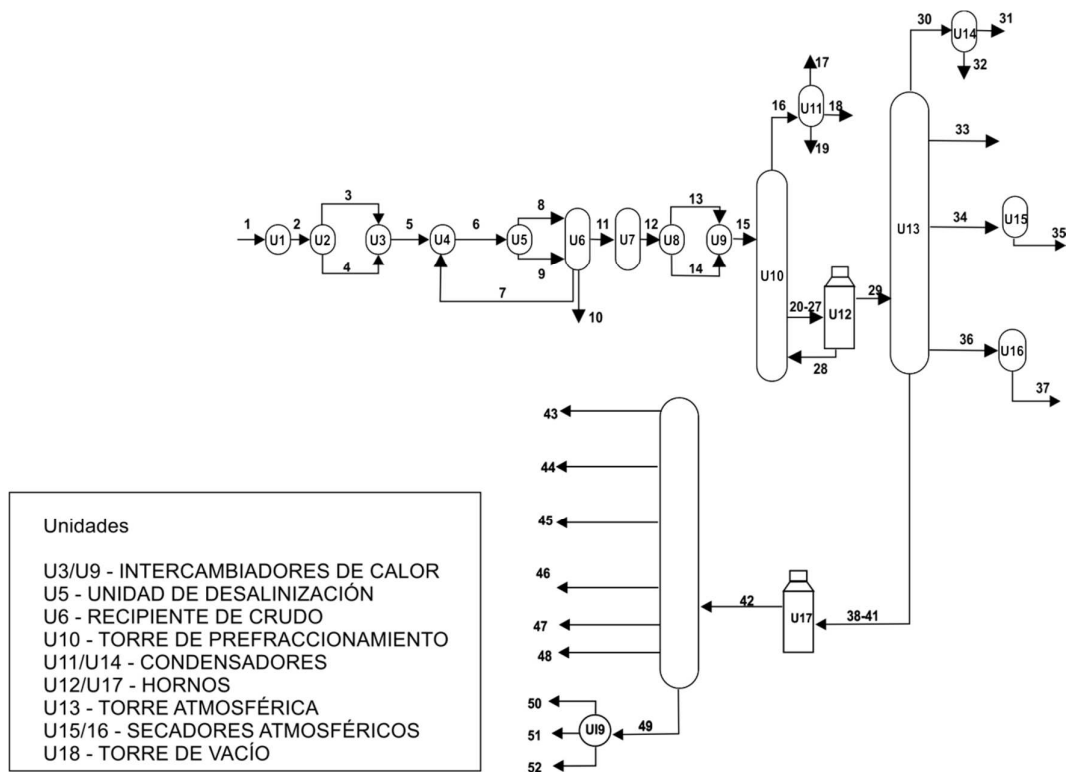


Figura 6.5. Diagrama de flujo – Caso 2

El diseño propuesto considera que un conjunto de 18 variables claves están sujetas a restricciones de estimabilidad, 9 de las cuales tienen además restricciones sobre la variabilidad. El grado de especificación para este ejemplo se muestra en la Tabla 6.7.

Caso 2	Especificaciones
	para las corrientes 2 5 15 29 31 32 38 39 40
$E_{le}^* \geq 1$	44 45 46 47 48 49 50 51
	52
$\sigma_{lp}^* \cdot 10^4$	$\sigma_{15}^* = 1.241$ $\sigma_{32}^* = 0.013$ $\sigma_{40}^* = 0.138$ $\sigma_{44}^* = 0.057$ $\sigma_{45}^* = 0.059$ $\sigma_{46}^* = 0.072$ $\sigma_{47}^* = 0.055$ $\sigma_{49}^* = 0.144$

Tabla 6.7. Especificaciones del diseño – Caso 2

En la Tabla 6.8 se exponen los resultados obtenidos por los diferentes algoritmos para 100 ejecuciones independientes de cada uno de ellos. El mejor valor encontrado corresponde a un valor de  $F$  igual a 1154.34 e implica medir las variables 10, 16, 31-33, 35, 37, 39-41, 43-48, 50-52. La precisión alcanzada sobre las variables claves, con esta propuesta de medición, es  $\hat{\sigma}_{15}=2510$ ,  $\hat{\sigma}_{31}=916$ ,  $\hat{\sigma}_{32}=84$ ,  $\hat{\sigma}_{40}=919$ ,  $\hat{\sigma}_{44}=379$ ,  $\hat{\sigma}_{45}=397$ ,  $\hat{\sigma}_{46}=478$ ,  $\hat{\sigma}_{47}=364$ ,  $\hat{\sigma}_{49}=566$ . Al igual que en el Caso 1, el mejor desempeño corresponde al algoritmo pPBIL-OS.

Metaheurística	Minimo	Media	Mediana	99 Percentil	Desvío	CV
pPBIL-OS	1154.34	1154.34	1154.34	1154.34	0	0
AffEDA	1197.38	1455.73	1429.93	1936.62	153.98	0.11
EBNA	2355.33	3907.56	3777.65	9736.83	1336.34	0.34

Tabla 6.8. Resultados obtenidos para el Caso 2 - pPBIL-OS, AffEDA y EBNA

En la Figura 6.6 se grafican, de manera ordenada, los mejores valores encontrados de la  $F$  en cada ejecución de los algoritmos en estudio. Este gráfico permite apreciar de manera clara cómo a medida que el tamaño del problema crece, la brecha entre los desempeños de las distintas técnicas de optimización comienza a ampliarse. Si bien, para el Caso 1 AffEDA aparecía como un buen competidor de pPBIL-OS, se observa que para un tamaño de instancia de 52 variables binarias, el procedimiento AffEDA ya no puede obtener el mínimo de referencia. En el caso del algoritmo EBNA, los resultados en general son pobres.

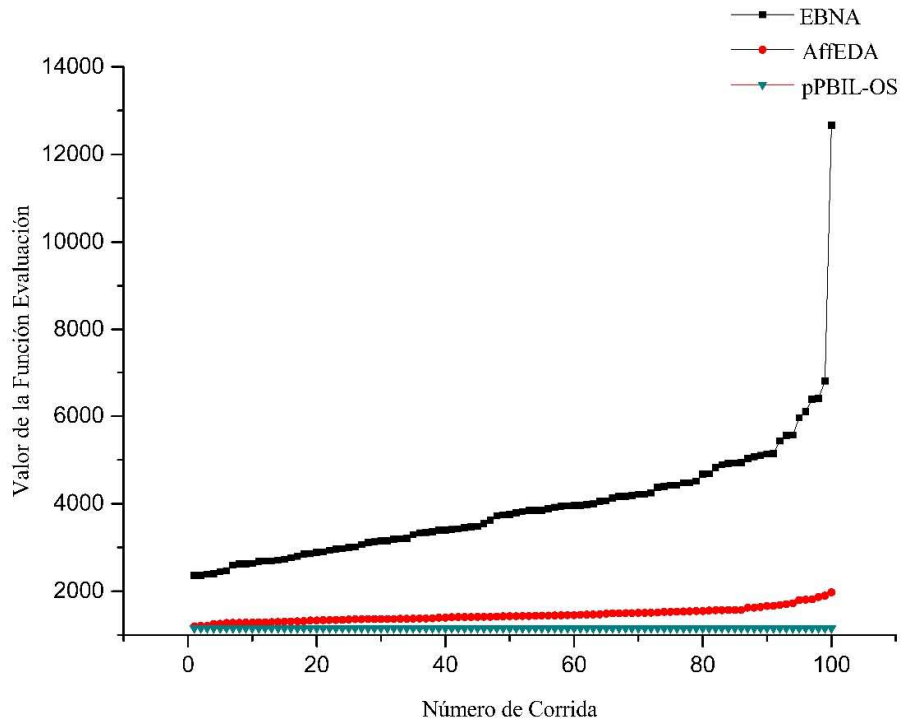


Figura 6.6. Mejores valores encontrados para el Caso 2 - pPBIL-OS, AffEDA y EBNA

A continuación se presentan los tiempos insumidos para la resolución de este diseño. Se realizaron 200 iteraciones en cada ejecución para cada uno de los algoritmos en ensayo. Los resultados se muestran en la Figura 6.7 mientras que el resumen de los estadísticos asociados se incluye en la Tabla 6.9.

La estrategia pPBIL-OS es la más demandante en cuanto al tiempo de ejecución. Sin embargo, debe tenerse en cuenta que en estas experiencias los tiempos reportados corresponden a una ejecución secuencial de cuatro PBIL-OS.



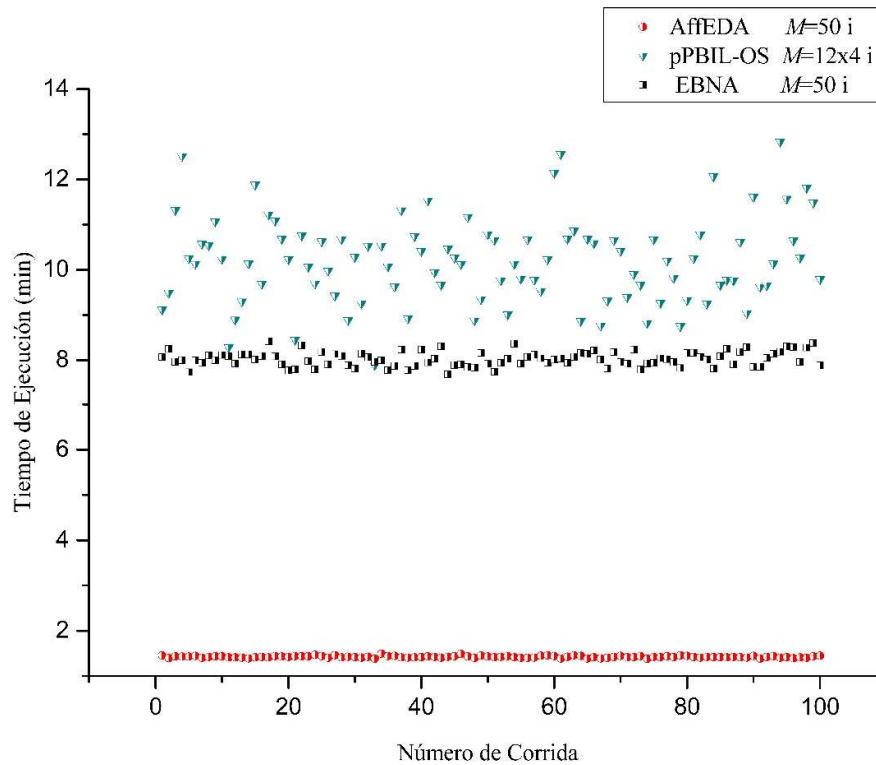


Figura 6.7. Tiempos de Ejecución para el Caso 2 - pPBIL-OS, AffEDA y EBNA

Metaheurística	Media	Desvío	Mínimo	Mediana	Máximo
pPBIL-OS	10.18	0.96	7.89	10.16	12.83
AffEDA	1.42	0.02	1.38	1.42	1.50
EBNA	8.02	0.16	7.68	8.01	8.41

Tabla 6.9. Estadísticos de los tiempos de ejecución - Caso 2

### 6.4 Caso 3

En este apartado se analizan los resultados obtenidos para el diseño óptimo de instrumentación en una planta de etileno simplificada que consta de 47 equipos y 82 corrientes. Este caso de estudio constituye la instancia de mayor envergadura considerada en esta tesis. Las técnicas de optimización propuestas tienen que ser capaces de encontrar soluciones óptimas o cuasi-óptimas entre un conjunto del orden de  $10^{24}$  posibles

combinaciones, cada una de ellas correspondiente a una propuesta de instrumentación diferente. Los datos correspondientes se muestran en la Tabla 6.10

Corriente	$C^m_i \cdot 10^4$	$\sigma_i \cdot 10^3$	$c_i \cdot 10^3$
1	4	1	4.6406
2	2.347	0.5867	2.723
3	6.347	1.5868	7.3634
4	6.347	1.5868	7.3634
5	1.556	0.3890	1.8053
6	7.903	1.9757	9.1685
7	7.903	1.9757	9.1685
8	7.903	1.9757	9.1685
9	7.921	1.9803	9.1894
10	7.921	1.9803	9.1894
11	7.903	1.9757	9.1685
12	7.921	1.9803	9.1894
13	7.921	1.9803	9.1894
14	6.725	1.6812	7.8019
15	1.178	0.2945	1.3668
16	7.903	1.9757	9.1685
17	6.798	1.6995	7.8866
18	6.449	1.6123	7.4817
19	6.429	1.6073	7.4585
20	6.449	1.6123	7.4817
21	$199.2 \cdot 10^{-4}$	0.0050	0.0233

Tabla 6.10. Datos – Caso 3

Corriente	$C^m_i \cdot 10^4$	$\sigma_i \cdot 10^3$	$c_i \cdot 10^3$
22	6.429	1.6073	7.4585
23	6.424	1.6060	7.4527
24	6.421	1.6053	7.4492
25	$6.53210^{-4}$	0.0002	0.001
26	6.421	1.6053	7.4492
27	3.589	0.8972	4.1638
28	2.832	0.7080	3.2856
29	3.589	0.8972	4.1638
30	1.952	0.4880	2.2647
31	1.637	0.4093	1.8993
32	4.47	1.1175	5.1859
33	1.952	0.4880	2.2647
34	0.9959	0.2490	1.1556
35	0.9561	0.2390	1.1094
36	1.191	0.2978	1.3819
37	0.8912	0.2228	1.0341
38	5.725	1.4312	6.6418
39	7.903	1.9757	9.1685
40	1.887	0.4718	2.1893
41	1.887	0.4718	2.1893
42	1.034	0.2585	1.1998
43	0.8535	0.2134	0.9904
44	0.0341	0.0085	0.0398
45	0.8194	0.2049	0.9508
46	0.0390	0.0098	0.0455
47	0.0732	0.0183	0.0852
48	1.0340	0.2585	1.1998
49	0.0732	0.0183	0.0852
50	0.6136	0.1534	0.712
51	0.6136	0.1534	0.712
52	0.6223	0.1556	0.7221

Tabla 6.10. Continuación: Datos - Caso 3

Corriente	$C^m_i \cdot 10^4$	$\sigma_i \cdot 10^3$	$c_i \cdot 10^3$
53	0.4113	0.1028	0.4774
54	0.3723	0.0931	0.4321
55	0.0087	0.0022	0.0103
56	0.6136	0.1534	0.712
57	0.6223	0.1556	0.7221
58	0.1483	0.0371	0.1723
59	5.577	1.3943	6.4701
60	3.218	0.8045	3.7334
61	2.359	0.5898	2.7369
62	0.0964	0.0241	0.1121
63	0.0518	0.0130	0.0604
64	0.0364	0.0091	0.0425
65	0.0154	0.0039	0.0181
66	1.952	0.4880	2.2647
67	2.359	0.5898	2.7369
68	0.0028	0.0007	0.0035
69	6.422	1.6055	7.4504
70	6.422	1.6055	7.4504
71	2.832	0.7080	3.2856
72	6.429	1.6073	7.4585
73	6.424	1.6060	7.4527
74	0.0445	0.0011	0.0054
75	0.6136	0.1534	0.712
76	0.6136	0.1534	0.712
77	0.0732	0.0183	0.0852
78	0.0732	0.0183	0.0851
79	0.6136	0.1534	0.712
80	0.0732	0.0183	0.0852
81	2.359	0.5898	2.7369
82	0.6136	0.1534	0.712

Tabla 6.10. Continuación: Datos - Caso 3

Las especificaciones del diseño para el Caso 3 abarcan un conjunto de 14 variables claves, 6 de ellas sujetas a restricciones de precisión. Un detalle de los requisitos sobre cada variable se presenta en la Tabla 6.11.

Caso 3	Especificaciones									
$E_{le} \geq 1$	para las corrientes	5	10	12	14	17	35	37	39	44
		56	62	69	70	77				
$\sigma_{lp}^* \cdot 10^3$	$\sigma_{10}^* = 1.584$	$\sigma_{17}^* = 1.359$	$\sigma_{35}^* = 0.200$	$\sigma_{39}^* = 1.580$	$\sigma_{56}^* = 0.123$	$\sigma_{69}^* = 1.284$				

Tabla 6.11. Especificaciones del diseño - Caso 3

El mejor valor encontrado o mínimo de referencia para el diseño analizado corresponde a un valor de  $F$  igual a 50845.16. La red de sensores asociada a dicho valor implica colocar caudalímetros máxicos en las siguientes corrientes: 1, 2, 5, 10, 12, 15, 21, 30, 33-37, 44, 50, 55, 56, 60, 62, 63, 66-68, 74-78, 82. Los desvíos estándares de las variables claves calculados con esta propuesta de medición son:  $\hat{\sigma}_{10} = 1.531 \cdot 10^3$ ,  $\hat{\sigma}_{17} = 0.985 \cdot 10^3$ ,  $\hat{\sigma}_{35} = 0.198 \cdot 10^3$ ,  $\hat{\sigma}_{39} = 1.166 \cdot 10^3$ ,  $\hat{\sigma}_{56} = 0.108 \cdot 10^3$ ,  $\hat{\sigma}_{69} = 0.988 \cdot 10^3$ . En la Tabla 6.12 se detallan los resultados obtenidos al aplicar las diferentes metaheurísticas para 100 ejecuciones de 200 iteraciones cada una.

Metaheurística	Minimo	Media	Mediana	99 Percentil	Desvío	CV
pPBIL-OS	50845.16	50886.63	50845.37	52909.78	412.88	0.0081
AffEDA	87282.49	104055.68	105128.89	121365.27	7142.19	0.0686
EBNA	105643.06	184736.07	155443.24	292404.24	67670.99	0.3663

Tabla 6.12. Resultados obtenidos para el Caso 3 - pPBIL-OS, AffEDA y EBNA

En dicha tabla se observa que el valor mínimo encontrado por el algoritmo EBNA es prácticamente el doble que el mínimo de referencia; además el CV (36.6 %) resulta entre uno y dos órdenes de magnitud superior con respecto al de las otras técnicas evolutivas. El desempeño mostrado por el algoritmo EBNA indica que éste, al menos con el conjunto de parámetros utilizados, no es apto para resolver una instancia de este tamaño.

Por su parte, el algoritmo AffEDA muestra una mejor repetitividad de resultados (CV= 6.8 %); sin embargo, el costo del valor mínimo obtenido supera en más del 70% al valor del mínimo de referencia.

La metodología pPBIL-OS proporciona el mejor valor encontrado. Los estadísticos calculados para dicho algoritmo son, en su totalidad, claramente superiores a los provistos por las otras técnicas de optimización. El 50 % de las veces pPBIL-OS obtiene resultados por debajo de 50845.47, es decir la mediana y el mínimo de referencia son casi idénticos; más aún, en el 99% de las ejecuciones el mejor valor alcanzado corresponde a un valor de  $F$  igual a 52909.78. Esta capacidad para replicar buenas soluciones en cada corrida se refleja también en un CV de sólo el 0.8%.

En la Figura 6.8 se grafican los valores mínimos de la  $F$  alcanzados en cada corrida, previo ordenamiento de los mismos. En ella puede observarse claramente el alejamiento respecto del mínimo de referencia de AffEDA y EBNA.

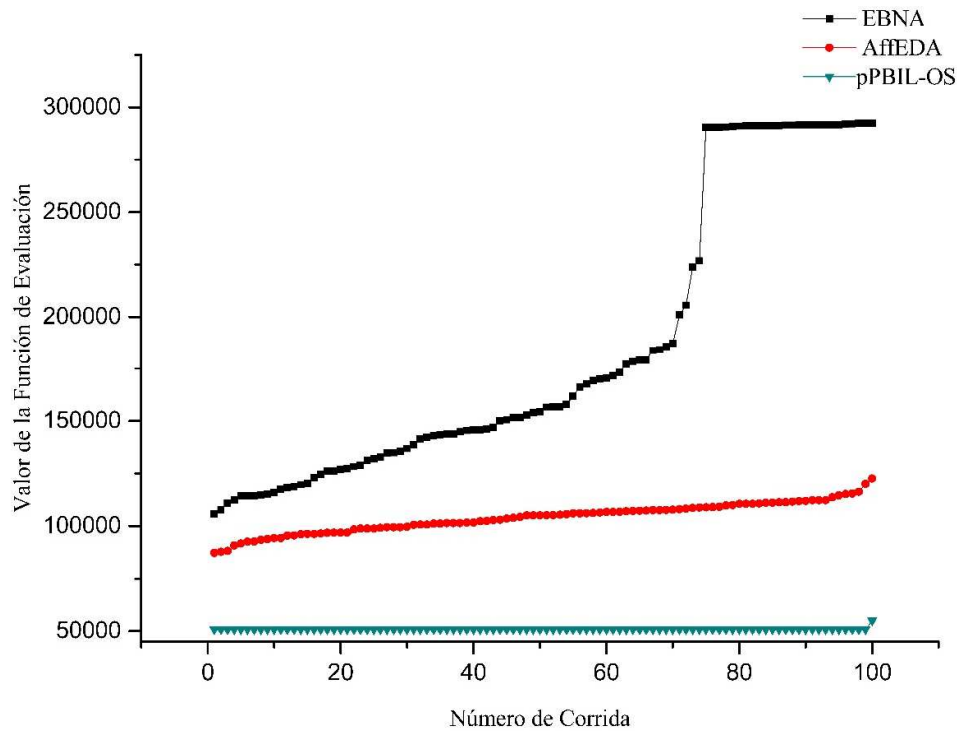


Figura 6.8. Mejores valores encontrados para el Caso 3 - pPBIL-OS, AffEDA y EBNA

En la Figura 6.9 se observan los tiempos de cómputo insumidos en cada ejecución por los diferentes algoritmos utilizados. A este respecto, las conclusiones son las mismas que las vertidas para el Caso 2. El algoritmo EBNA es el que más tiempo consume, aún cuando las poblaciones que evolucionan son relativamente pequeñas. En la Tabla 6.13 se muestran los estadísticos correspondientes.

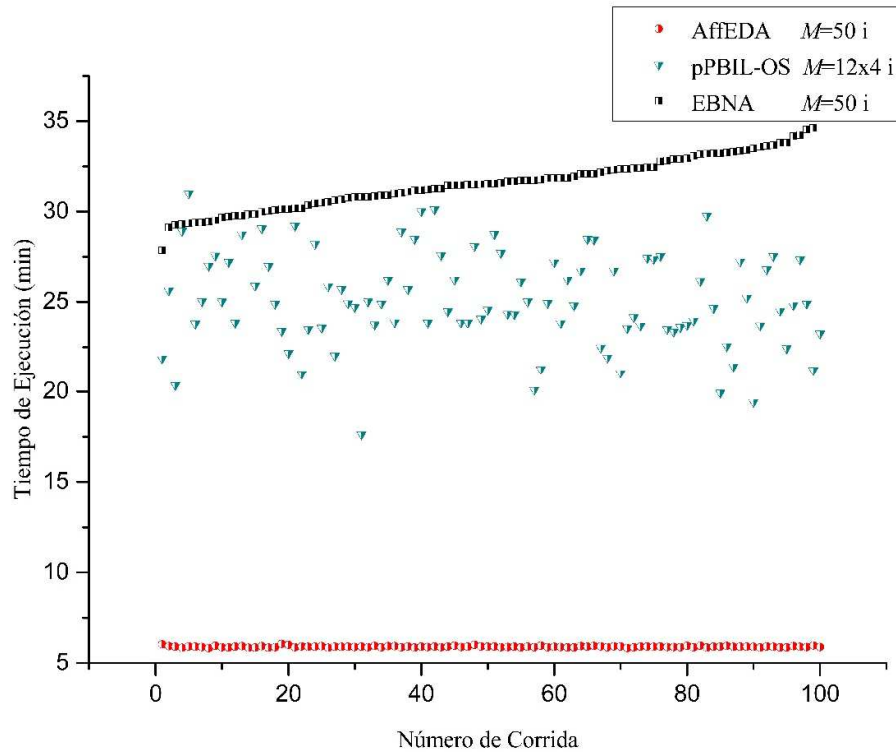


Figura 6.9. Tiempos de Ejecución para el Caso 3 - pPBIL-OS, AffEDA y EBNA

Metaheurística	Media	Desvío	Mínimo	Mediana	Máximo
pPBIL-OS	25.13	2.66	17.62	24.90	30.99
AffEDA	5.90	0.04	5.82	5.90	6.07
EBNA	31.60	1.47	27.86	31.54	35.61

Tabla 6.13. Estadísticos de los tiempos de ejecución - Caso 3

Para esta instancia, Caso 3, en particular se realizaron ejecuciones adicionales con el objetivo de determinar el impacto del tamaño de la población en el desempeño del procedimiento AffEDA, ya que este algoritmo con aprendizaje estructural es, a diferencia del EBNA, el que parece ofrecer posibilidades de generar soluciones más cercanas al mínimo de referencia. Las experiencias consistieron en realizar nuevamente 100 ejecuciones con poblaciones formadas por 50, 100 y 300 individuos. Los valores



ordenados de  $F$ , incluyendo aquellos obtenidos con pPBIL-OS, se presentan en la Figura 6.10.

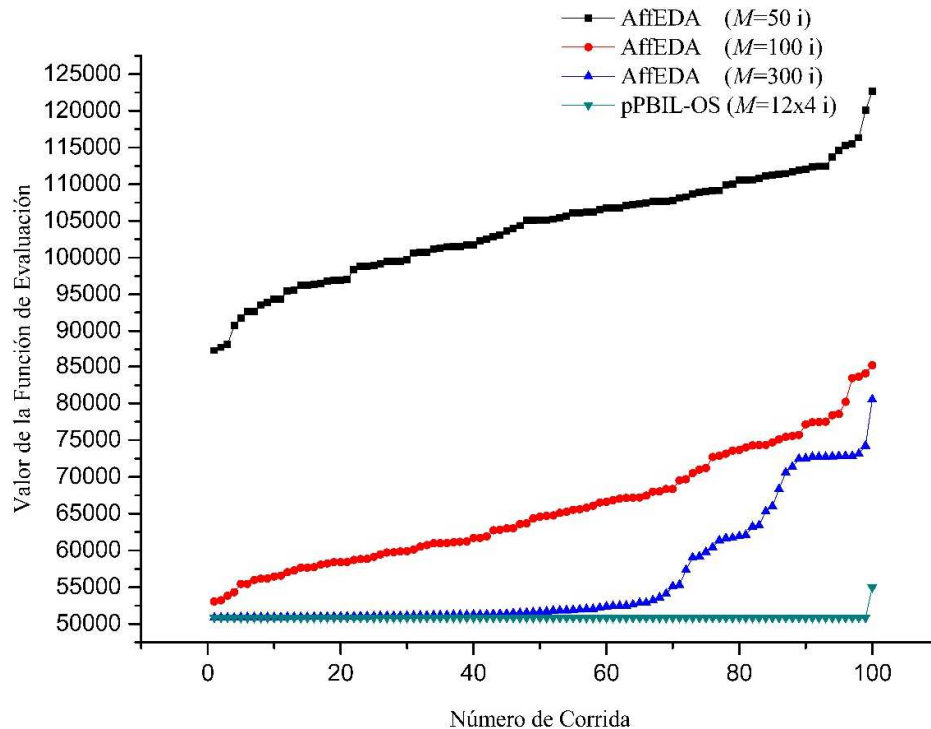


Figura 6.10. Mejores valores encontrados para el Caso 3 - pPBIL-OS, AffEDA

En dicha figura se observa que AffEDA no puede encontrar el valor del mínimo de referencia en ninguna de las ejecuciones para poblaciones constituidas por 50 y 100 individuos. El algoritmo comienza mejorar su desempeño cuando el tamaño de la misma aumenta a 300 individuos. Sin embargo, esto implica sextuplicar el tamaño de la población respecto de pPBIL-OS, con el consiguiente aumento en la cantidad de veces que se calcula la  $F$ . Aún ampliando considerablemente el conjunto de potenciales soluciones a explorar, AffEDA sólo puede replicar el mínimo de referencia en el 50 % de las corridas, obteniendo incluso valores de la  $F$  por encima de rango intercuartílico, tal como puede apreciarse en los diagramas de caja y bigote de la Figura 6.11. Por su parte,

pPBIL-OS muestra un comportamiento casi ideal, ya que converge al valor óptimo o mínimo de referencia prácticamente en el 100% de las corridas.

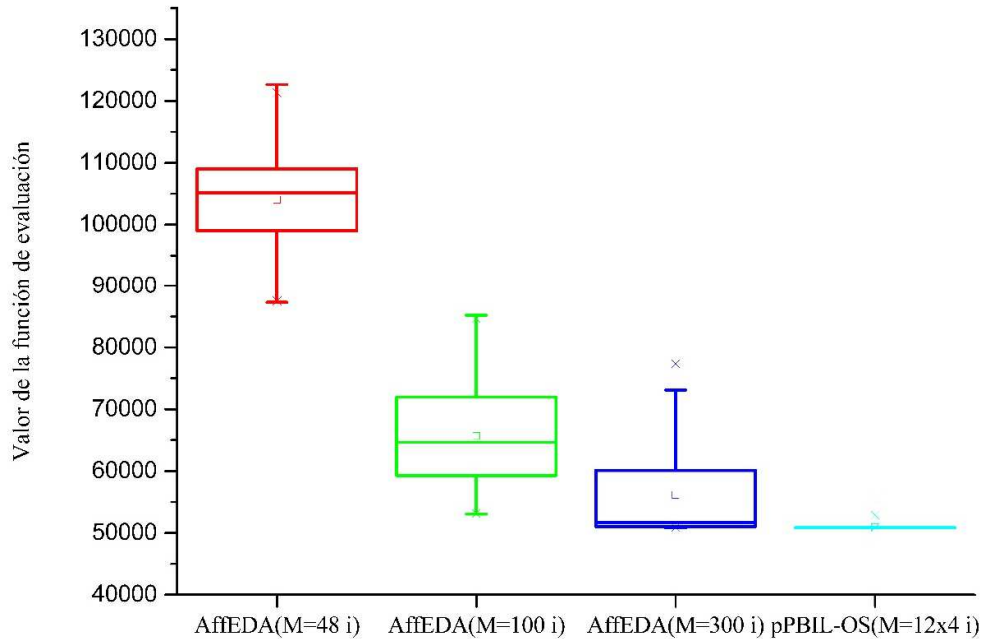


Figura 6.11. Estadísticos de la función de evaluación – Caso 3 - pPBIL-OS y AffEDA

## 6.5 Conclusiones

En este capítulo se muestran los resultados de diferentes experiencias realizadas para un conjunto de PDRS. Dicho conjunto abarca diferentes grados de especificaciones de diseño e instancias de tamaño creciente del problema de optimización. En este sentido se han considerado tres procesos que involucran: 28 corrientes y 11 equipos, 52 corrientes y 19 unidades, y por último 82 corrientes y 47 equipos.

Las experiencias se planificaron con el objetivo de comparar el desempeño de tres AEDs de diferente orden. Por una parte, los procedimientos AffEDA y EBNA son estrategias multivariadas, dotadas de diferentes técnicas de aprendizaje estructural, que

intentan modelar la interrelación entre las variables del problema. Por la otra, la metodología pPBIL-OS es un algoritmo univariado, que supone independencia entre las variables, pero que está hibridizado con una heurística de búsqueda local.

El algoritmo EBNA emplea un aprendizaje estructural de las dependencias existentes entre cualquier conjunto de variables basado en RB. Los resultados de las experiencias realizadas indican que dicho procedimiento no resulta eficiente para la resolución de los problemas de diseño de instrumentación planteados. La calidad de las soluciones obtenidas resultó muy pobre en todos los casos, y además consumió elevados tiempos de cómputo. Se investigó el efecto del aumento del tamaño de la población sobre el desempeño del algoritmo para la instancia más pequeña (28 variables), pero no se observaron mejorar sustancialmente. Este análisis no pudo realizarse para las instancias mayores dado el excesivo tiempo que insumía.

El enfoque utilizado por el procedimiento AffEDA para simular la relación entre las variables intervinientes difiere del anterior, ya que supone dependencia dentro de conjuntos disjuntos de variables. La búsqueda de la partición que mejor represente los datos muestreados en cada iteración y la estimación de los parámetros constituyen la esencia del método. En líneas generales, el desempeño del algoritmo AffEDA es mejor que el del EBNA. En el caso de instancias de tamaño pequeño (28 variables), los valores mínimos alcanzados son iguales a los mínimos de referencia, aunque no puede converger a estos valores en todas las ejecuciones realizadas. A medida que el número de variables crece, el procedimiento AffEDA no logra encontrar soluciones de alta calidad y queda atrapado en subóptimos con muy baja diversidad. Esta dificultad puede resolverse parcialmente aumentando el tamaño de la población, ya que a medida que se amplía la capacidad exploratoria del algoritmo mejora la calidad de las soluciones encontradas.

El algoritmo pPBIL-OS aparece como la metodología más eficaz en todos los casos probados. Puede hallar y replicar los mejores valores encontrados en la mayoría de las corridas. En los problemas pequeños y de mediana escala, el mínimo de referencia coincide con la media, y el CV es igual a cero, calculado en base a 100 corridas. En el problema de mayor envergadura, si bien el CV aumenta ligeramente, el algoritmo es aún capaz de hallar soluciones de muy bajo costo.

De los resultados observados se desprende que para el PDRS los dos enfoques más exitosos fueron pPBIL-OS y AffEDA con grandes poblaciones, estrictamente en dicho orden. El primero de los algoritmos trabaja con tamaños de población pequeños, lo que implica una exploración reducida del espacio de búsqueda, y cuenta con un mecanismo de búsqueda local que juega un rol fundamental. En el algoritmo pPBIL-OS se ha privilegiado la explotación de buenas regiones en detrimento de la exploración de grandes poblaciones.

## 6.6 Nomenclatura

$C^m_i$	Valor de caudal verdadero para la corriente $i$
$c_i$	Costos del caudalímetro $i$ .
$E_{le}$	Grado de estimabilidad de la variable $le$
$F$	Función de evaluación
$M$	Número de individuos en una población dada
$NPBIL$	Número de instancias independientes del algoritmo PBIL

Letras Griegas

$\sigma_i$	Desvío estándar de la $i$ -ésima medición
$\hat{\sigma}_{lp}$	Desvío estándar de la estimación de la $lp$ -ésima variable
$\sigma_{lp}^*$	Desvío estándar superior de la estimación de la $lp$ -ésima variable

## 6.7 Acrónimos

AED	Algoritmo de Estimación de Distribuciones
AffEDA	<i>Affinity Propagation Estimation Distribution Algorithm</i>
EBNA	<i>Estimation of Bayesian Network Algorithm</i>
pPBIL	Algoritmo PBIL paralelo
PDRS	Problema de Diseño de Redes de Sensores
OS	Oscilación Estratégica
RB	Redes Bayesianas

## 6.8 Referencias

Carnero, M., Hernández, J., Sánchez, M. Optimal Sensor Location in Chemical Plants using Estimation of Distribution Algorithms. Artículo en revisión, enviado a *ISA Transaction*, Diciembre de 2015.

- Etxeberria, R., Larrañaga, P. Global optimization with Bayesian networks. *Special Session on Distributions and Evolutionary Optimization*, 332-339. La Habana, Cuba. II Symposium on Artificial Intelligence, CIMA99, (1999).
- Gala, M., Bagajewicz, M. Rigorous Methodology for the Design and Upgrade of Sensor Networks using Cutsets *Industrial and Engineering Chemistry Research*. **45**, N 21, 6679-6686, (2006).
- Nguyen, D.Q., Bagajewicz, M. Parallel Computing Approaches to Sensor Network Design Using the Value Paradigm. *Computers & Chemical Engineering*, **35** (6), 1119-1134, (2011).
- Santana, R.; Larrañaga, P.; Lozano, J. Learning Factorizations in Estimation of Distribution Algorithms Using Affinity Propagation. *Evolutionary Computation* **18**(4), 515–546, (2010).

# ***Capítulo 7***

**Conclusiones y Trabajos Futuros**

## 7.1 Conclusiones

En esta tesis se han desarrollado e implementado metodologías de resolución para el Problema de Diseño Óptimo de Redes de Sensores, PDRS, en plantas de proceso cuya operación se representa mediante sistemas linealizados de ecuaciones algebraicas.

Las estrategias propuestas emplean motores de optimización híbridos con el fin de explotar las ventajas de los algoritmos estocásticos de búsqueda global y local. Dichas metodologías, si bien no garantizan optimalidad, ofrecen buenas soluciones en tiempos que no crecen exponencialmente con el tamaño de la entrada. Además son lo suficientemente flexibles como para ser adaptadas a otras formulaciones del PDRS.

Inicialmente se investigó la resolución del diseño empleando algoritmos de búsqueda local inspirados en la heurística Búsqueda Tabú (BT). En tal sentido se desarrollaron los procedimientos de diseño: Búsqueda Tabú Clásica y Búsqueda Tabú enriquecida con la técnica de Reencadenamiento de Caminos. Se analizó el desempeño de las mismas para la resolución de dos casos de estudio (28 corrientes y 11 equipos) que involucran restricciones de diferente complejidad. Se observó que cuando la complejidad aumenta, se reduce la robustez de los algoritmos, ya que éstos presentan dificultades para reproducir el mejor valor encontrado cuando el punto inicial se modifica.

También se estudió el comportamiento de un algoritmo ad-hoc de diseño de instrumentación basado en la Búsqueda Dispersa. Este fue el primer procedimiento híbrido presentado en esta tesis, dado que en él se combinan métodos de búsqueda global con técnicas de inspección por entornos diferentes a las empleadas en la BT. Se obtuvo una mejor reproducibilidad de los resultados para el caso de estudio que comprende restricciones más complejas en relación con los alcanzados con las estrategias previamente citadas.



Con respecto a los métodos de optimización basados en poblaciones, métodos globales, se investigó la aplicabilidad de los Algoritmos de Estimación de Distribuciones (AEDs) en la resolución del PDRS. Se utilizaron distintos exponentes de AEDs, en orden creciente de complejidad, con la finalidad de analizar el impacto de los diferentes modelos probabilísticos sobre las soluciones obtenidas. Cabe mencionar que los AEDs empleados no tenían incorporado ningún conocimiento específico del problema, partían de poblaciones inicializadas al azar y no estaban hibridizados con ningún mecanismo de búsqueda local. Aún así evidenciaron un desempeño aceptable ya que fueron capaces de hallar buenas soluciones. Se observó que todos presentaron dificultades para resolver el caso de estudio más complejo. Además se concluyó que los inconvenientes adicionales que implica considerar un modelo gráfico probabilístico de mayor orden, ya sea por los mayores tiempos de cómputo o por el hecho de sumar un problema de optimización adicional, como en el caso de aprendizaje con redes bayesianas, no implica una mejora sustancial en la calidad y repetitividad de las mejores soluciones.

El análisis crítico de las ventajas y desventajas propias de las estrategias de búsqueda local, global e híbridas condujo al desarrollo de una nueva metodología de resolución del PDRS, denominada pPBIL-OS (*Parallel Population Based Incremental Learning – Strategic Oscillations*). Esta consiste en un esquema de resolución híbrido que combina las ventajas de los algoritmos poblacionales y de los que aplican búsquedas por entorno. La estrategia maestra es un AED de orden uno (PBIL) que es capaz de manejar simultáneamente un conjunto de soluciones, mientras que la técnica subordinada es la heurística OS, que trabaja en el marco de la BT. El nuevo procedimiento emplea un número predeterminado de subpoblaciones que evolucionan de manera independiente, pudiendo intercambiar información entre ellas con cierta probabilidad. En cada una de estas subpoblaciones actúan de manera controlada mecanismos de búsqueda local que

aseguran una inspección más sesgada hacia zonas promisorias del espacio de soluciones. El comportamiento de pPBIL-OS para la resolución de los dos casos de estudio resultó notablemente superior al de las metodologías previamente mencionadas. A pesar de su naturaleza probabilística, el conjunto de las soluciones obtenidas en diferentes corridas del procedimiento muestra una dispersión igual a cero y una convergencia hacia buenas soluciones.

Con el objetivo de investigar la escalabilidad del método pPBIL-OS, se resolvieron además los problemas de diseño correspondientes a una unidad de destilación de crudo, que consta de 52 corrientes y 19 unidades, y a una planta de producción de etileno simplificada, que comprende 82 corrientes y 47 equipos. En todos los casos estudiados pPBIL-OS mostró gran capacidad para hallar soluciones de alta calidad, así como una buena convergencia hacia estos valores en cada ejecución del mismo.

Las principales ventajas del método propuesto para la resolución de diseños de redes de instrumentos son las siguientes:

- a) No requiere de una formulación matemática especial de la función objetivo, más aún, está diseñado para utilizar las funciones objetivo como cajas negras, por lo tanto puede adaptarse a la resolución de diferentes diseños de instrumentación, incluyendo diseños con criterios múltiples de desempeño.
- b) Explora en forma simultánea diferentes regiones del espacio de búsqueda, a la vez que puede intensificar la inspección de algunos entornos promisorios. En cualquier momento de la evolución del algoritmo se dispone de un conjunto de candidatos o potenciales soluciones, en contraste con los métodos secuenciales o constructivos que consideran una única solución o una parte de ella, respectivamente. Esta característica permite la obtención de más de una

solución de alta calidad en cada ejecución.

Los tiempos de cómputo asociados crecen naturalmente con el tamaño del problema a resolver. Si bien este aspecto no es tan crítico como en los algoritmos exactos, constituye un verdadero desafío el poder reducirlos con el objetivo de abordar el diseño de redes de sensores en problemas de cientos o miles de variables. Una contribución en este sentido es la ejecución del algoritmo en una arquitectura en paralelo. Este tema se discute en la sección siguiente.

## **7.2 Trabajos futuros**

Los algoritmos descritos en esta tesis han sido desarrollados bajo un paradigma de programación de tipo secuencial. Esto es, las instrucciones que conforman el programa de computación que implementa el algoritmo se cumplen una tras otra. De esta manera, el tiempo necesario para la resolución de una instancia determinada depende de la optimización del código secuencial utilizado, y está limitado por dos factores: la capacidad de ejecución de instrucciones y la velocidad de intercambio entre memoria y CPU.

Sin embargo, la búsqueda de un mayor rendimiento en las aplicaciones, ha conducido a la división de las labores de cómputo en más de un procesador, cada uno de los cuales puede desempeñar tareas de menor complejidad y que son susceptibles de ser resueltas mediante la ejecución simultánea de instrucciones. Esta alternativa, consistente en el empleo de varios procesadores que cooperan de alguna manera para resolver un problema, recibe el nombre de computación paralela.

El problema se divide en subproblemas, cada uno de los cuales es resuelto por procesadores diferentes de manera simultánea. Por supuesto, deberá haber una

comunicación entre los procesadores para intercambiar resultados parciales. Debido al tiempo utilizado en esta comunicación, un número  $nproc$  de procesadores independientes no alcanzan la solución de un problema en una fracción  $nproc$  del tiempo que le hubiera llevado a un procesador secuencial.

A pesar de ello, el cómputo paralelo permite la consecución de mejores rendimientos en diferentes tipos de problema. En general, las heurísticas desarrolladas en esta tesis son paralelizables y, en particular el algoritmo pPBIL-OS es “naturalmente” paralelizable.

Se han propuesto diferentes arquitecturas en computación paralela. Difieren en la forma de acceder a la memoria por parte de los procesadores, u operar de manera síncrona o asíncrona, o ejecutar el mismo algoritmo en cada subproblema o no. En una primera categoría se consideran los sistemas multiprocesadores con arquitectura de memoria compartida. En este caso la comunicación entre procesadores se da a través de la lectura y/o escritura en la memoria común. Otro tipo de esquemas son los sistemas multicomputadoras con memoria distribuida, es decir, no compartida. En este caso los procesadores se comunican a través del pasaje de mensajes.

Un *cluster* es un conjunto de computadoras independientes integradas en una única unidad de cómputo. La posibilidad de integrar distintos tipos de *hardware*, de propósito general e incluso de bajo costo comparativo, ha permitido a los *clusters* ubicarse como una de las arquitecturas más difundidas para la computación paralela en la actualidad.

Más aún, el advenimiento de procesadores multinúcleo, los cuales integran más de una CPU en un mismo chip, no sólo ha cambiado la estrategia en el desarrollo del software sino que claramente aporta un beneficio adicional al desempeño del *cluster*. Éste queda ahora formado por diferentes unidades con varios procesadores y capacidad de cómputo

paralelo en cada una de ellas, interactuando en paralelo entre sí. En otras palabras, puede hablarse de paralelismo en dos niveles: interno (entre los núcleos de cada procesador multinúcleo) y externo (entre los procesadores del *cluster*).

Resulta claro que esta enorme ventaja que proporciona la arquitectura descrita tiene un costo: una mayor complejidad en la programación paralela, respecto de la secuencial clásica. Sin embargo, existen herramientas de software encargadas de facilitar el desarrollo de aplicaciones paralelas, que se agrupan bajo el nombre genérico de *middleware*.

*Middleware* es software diseñado para ocultar al programador algunos aspectos del cómputo paralelo y balance de carga, conectar aplicaciones y permitir el intercambio de datos entre ellas.

En rigor, se debe hacer la distinción entre procesamiento paralelo y procesamiento distribuido. El primero implica que los procesadores involucrados comparten la memoria, mientras que en el segundo caso, cada procesador tiene su propia memoria. Cuando se hace referencia a paralelizar una metaheurística se consideran aquí ambas alternativas.

En lo que sigue se propone la implementación de algoritmos paralelos para dar solución a los problemas planteados en esta tesis, con el objetivo de disminuir al máximo posible los tiempos de cómputo asociados.

Las metaheurísticas estudiadas en esta tesis tienen la característica de ser susceptibles de paralelización con relativamente bajos tiempos de intercambio de información entre procesadores. La implementación de los algoritmos aquí descriptos permite la evolución cuasi independiente en diferentes equipos, con períodos cortos de intercambio de información entre ellos. Es por eso que la utilización de agrupamiento de

computadoras (*clusters*) es una buena alternativa, considerando además el bajo costo relativo de adquisición.

Se prevé la adquisición de computadoras equipadas con varios procesadores e interconectarlos sobre una red de alta velocidad, para posteriormente adaptar los algoritmos desarrollados con el fin de trabajar de manera paralela sobre el *hardware* disponible. Deberán tenerse en cuenta algunos aspectos importantes para realizar la implementación:

- a) Arquitectura. Aunque no de manera excluyente, se propone trabajar sobre un *cluster* de computadoras interconectadas a través de una red *Ethernet* de alta velocidad.
- b) Estudio y comparación de diferentes estrategias de paralelización de metaheurísticas. En ese sentido se investigarán, entre otras, alternativas tales como: Maestro-esclavo, modelo de islas y estructuras celulares.
- c) Diseño e implementación de las extensiones de los algoritmos para realizar la programación paralela. Incluye la selección de las herramientas de software adecuadas y el estudio de los mecanismos de sincronización de tareas.
- d) Clasificación de las metaheurísticas. La descentralización dependerá de la forma en que se inspecciona el espacio de búsqueda. Puede existir una sola solución que se va mejorando en cada iteración o bien una población que evoluciona.

Se espera lograr, a partir de la implementación de las extensiones paralelas de los algoritmos, la resolución de instancias de mayor tamaño de los problemas planteados, así como la obtención de un paquete de software de optimización combinatoria capaz de resolver otros problemas dentro del campo de la Ingeniería.

### 7.3 Acrónimos

BT            Búsqueda Tabú

AED            Algoritmo de Estimación de Distribuciones

PBIL            *Population Based Incremental Learning*

pPBIL-OS      *Parallel Population Based Incremental Learning – Strategic  
Oscillations*

PDRS            Problema de Diseño de Redes de Sensores

OS            Oscilación Estratégica