



UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO

PROGRAMA DE POSGRADO EN CIENCIAS DE LA TIERRA

IMPLEMENTACIÓN DE UN SISTEMA DE
MANEJO ÓPTIMO DE RECURSOS NATURALES
(SMORN) . APLICACIÓN EN AGUA SUBTERRÁNEA

T E S I S

QUE COMO REQUISITO PARA OBTENER EL GRADO DE:
MAESTRO EN CIENCIAS DE LA TIERRA

P R E S E N T A

OLIVER XAVIER LÓPEZ CORONA

JURADO EXAMINADOR:

DRA. GRACIELA HERRERA (PRESIDENTE)

DR. OSCAR ESCOLERO (VOCAL)

DR. PABLO PADILLA (SECRETARIO)

DRA. MARÍA DEL CONSUELO BONFIL (SUPLENTE)

DR. OCTAVIO PÉREZ MAQUEO (SUPLENTE)

DIRECTOR DE TESIS: DR. PABLO PADILLA



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

Quiero agradecer a mi esposa Elvia por ser siempre mi inspiración para ser una mejor persona. Gracias por no dejarme perder el rumbo, por tu amor y comprensión.

A mis padres, hermano y familia política, quienes en todo momento me han apoyado con su amor y cariño.

Al Dr. Pablo Padilla por haberme dado la oportunidad de trabajar con él, por su apoyo en un momento crucial de mi desarrollo académico, por la gran confianza que en todo momento deposito en mi y por haber compartido sus ideas conmigo, ideas que en ultima instancia están marcando para mi una linea de desarrollo profesional.

Al Dr. Oscar Escolero por haber sido desde el inicio del posgrado un maestro ejemplar, una guía académica y ejemplo de investigador. Por haber despertado en mi un profundo interés en la hidrología y su gestión. Por su confianza y guía.

Al Dr. Sergio Mendoza por su apoyo, por compartir conmigo experiencias y recursos computacionales, por ser siempre un ejemplo a seguir y por todos los consejos que de forma tan abierta siempre ha compartido conmigo. Pero sobre todo, gracias por su amistad.

A mis sinodales, por su paciencia, apoyo y recomendaciones, que hicieron de éste un mejor trabajo.

A la Dra. Christina Siebe por haber sido una maestra tan comprometida, por haber despertado en mi al edafólogo que no sabía tenía dentro.

A mi compañero Felipe Armas que de forma tan generosa me dejo trabajar con su modelo.

Por su puesto gracias al CONACyT quien hizo posible este proyecto atreves de su apoyo económico.

Gracias finalmente a la UNAM por ser para mi y muchos otros, una segunda casa. Espero sinceramente que este trabajo sea digno de esta institución.... GOYA, GOYA....!!

Índice general

Prefacio	1
Resumen	2
1. Introducción	4
§1.1. Estructura de la Tesis.	4
§1.2. Agua y desarrollo sustentable.	5
§1.2.1. Recursos hídricos en crisis.	5
§1.2.2. Importancia de estudiar el agua subterránea	10
2. Teoría matemática de control óptimo	13
§2.1. Semblanza Histórica	13
§2.2. Conceptos fundamentales.	15
§2.3. Teoría clásica de control.	16
§2.4. Teoría moderna de control.	24
§2.4.1. Formulación del problema básico de control óptimo.	26
§2.4.2. Ejemplos de control óptimo básico.	27
§2.4.3. Controlabilidad.	31
§2.5. Ejemplos de control en Ciencias de la Tierra.	41
3. Control óptimo en agua subterránea	44
§3.1. Modelación del agua subterránea	44
§3.1.1. Mecánica de los medios deformables.	45
§3.2. El flujo subterráneo.	50
§3.2.1. Solución numérica	53
§3.3. Manejo óptimo.	55
§3.3.1. Programación Lineal (PL)	56
§3.3.2. Programación No Lineal (PNL)	57
§3.3.3. Programación Dinámica (PD)	59

§3.4. Computo Evolutivo.	62
§3.4.1. Algoritmos Genéticos	63
§3.4.2. Optimización de un función simple usando AG	65
§3.4.3. Representación	66
§3.4.4. Población inicial.	67
§3.4.5. Evaluación de la Adecuación	67
§3.4.6. Operadores genéticos	68
§3.4.7. Parámetros	69
§3.4.8. Algunas aplicaciones en agua subterránea	69
4. SMORN y su aplicación en agua subterránea.	71
§4.1. Objetivos	71
§4.2. Estructura de SMORN	72
§4.2.1. El código de optimización.	74
§4.3. Sitio de estudio.	83
§4.3.1. Hidrología de la cuenca.	85
§4.3.2. Hidrogeología	86
§4.3.3. Geología	87
§4.3.4. Suelos y Clima	87
§4.4. Modelo de la cuenca del Río Duero.	89
§4.5. Resultados	90
§4.6. Discusión	95
§4.7. Conclusiones	100
§4.8. Perspectivas futuras.	101
Apéndices	104
Línea de tiempo	104
GNU Free Documentation Licence	105
Códigos fuente SMORN	113
Bibliografía	151

Prefacio

El diseño del presente trabajo es una adaptación a LyX del original escrito en LaTeX por el Dr. Sergio Mendoza ([sergio www.mendoza.org/sergio](http://www.mendoza.org/sergio)). Este trabajo fue realizado usando únicamente software libre. En conformidad con las ideas de acceso libre a la cultura, el autor desea expresar su consentimiento para utilizar esta obra de acuerdo con la siguiente licencia.

Copyright © 2009 Oliver López mail <lopezoliverx@gmail.com>. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU (GNU Free Documentation Licence), versión 1.2 o cualquier otra versión posterior publicada por la Fundación de Software Libre (Free Software Foundation); con las siguientes secciones invariantes: Resultados. Una copia de la licencia se incluye en la sección Apéndice bajo el título “GNU Free Documentation Licence” la cual puede ser directamente consultada en FDL www.gnu.org/licenses/licenses.html#FDL

Para mayor información sobre la Fundación de Software Libre consultar fsf www.fsf.org y para el proyecto GNU gnu www.gnu.org



Solo como Gnu

nómada pensamiento

se libre siempre

Resumen

Hoy en día, el agua subterránea es el recurso natural mas extraído del mundo, aportando alrededor del 50 % del abastecimiento de agua potable en el mundo, 40 % de la usada en la industria y 20 % de la usada en agricultura. Se entiende entonces que el agua subterránea es un recurso estratégico de cualquier país, por lo que es imperante contar con herramientas para diseñar planes de manejo sustentable, que coadyuven al aseguramiento del recurso. El presente trabajo de tesis tiene justamente ese objetivo e inclusive va más allá, al proponer una metodología que podría ser aplicable, en principio, a cualquier recurso natural. Esta metodología fundamentada en la teoría matemática del control óptimo; considera además que la mayoría de problemas de gestión de recursos naturales en la actualidad, son problemas multiobjetivo. Por lo anterior se uso el paradigma del computo evolutivo, pues se muestra que tiene un numero importante de ventajas frente a métodos tradicionales como la Programación No Lineal. Con estas ideas en mente, se realizó una implementación computacional basado en el GAToolbox, un código de optimización por Algoritmos Genéticos, libre para aplicaciones académicas y altamente modular, escrito por Kumara Sastry. Acoplado esta implementación a un modelo en MODFLOW del acuífero del Río Duero en Michocacán, provisto por el M. en C. Felipe Armas; se resolvió un problema de tres funciones objetivo: maximizar el total de agua extraída, minimizar el abatimiento promedio y minimizar su velocidad promedio. Los resultados muestran que el sistema converge a cuatro tipos diferentes de soluciones óptimas: una que privilegia la extracción, una que privilegia la conservación y otras dos que corresponden a casos intermedios, donde la extracción y conservación están equilibradas. Estas soluciones, al ser parte de un frente de pareto, son completamente equivalentes desde la óptica de la optimización multiobjetivo, por lo que se integro un análisis post-optimización usando Teoría de Juegos. Para ello, se construyo un juego extendido entre tres jugadores, la autoridad encargada del manejo del acuífero, los usuarios del mismo y un sector de la sociedad cuyo mayor interés es la protección del acuífero. Al calcular los Equilibrios de Nash Asociados, se encontró que el tipo de solución que debe adoptarse es una de las dos en las que extracción y conservación están equilibradas. De esta forma se muestra, que el sistema propuesto no solo es capaz de generar políticas de manejo óptimo, sino que

además constituye una herramienta valiosa en el proceso de toma de decisiones y resolución de conflictos. Así mismo, se esbozo una futura aplicación en el manejo suelo, en la cual se busca evaluar cual sería el mejor sitio para poner un nuevo basurero municipal en Tuxpan, Michoacán, considerando las propiedades geomorfo-pedológicas, costos e impactos ambientales. Se muestra que los resultados parciales obtenidos al momento están en buen acuerdo con estudios de campo independientes.

Now days, groundwater is the most extracted natural resource in the world, contributing 50 % of potable water in the world, 40 % of the required in industry and 20 % in agriculture. It is then understood that groundwater is a strategic resource of any country, so it is prevalent have tools to design sustainable management plans which help to safeguard the resource. This thesis work has precisely that goal and even goes further by proposing a methodology that could be applicable in principle to any natural resource. This methodology based on the mathematical theory of optimal control; also believes that most problems of natural resource management at present, are multi-objective kind of problems. For that reason. the paradigm of evolutionary computation was selected, because it has showed a number of important advantages over traditional methods such as nonlinear programming. With these thoughts in mind, we conducted a computational implementation based on GAtoolbox optimization code written by Kumara Sastry. We coupled this optimization engine to MODFLOW aquifer model from Michocacán Duero River, provided by the Msc. Felipe Armas, student of Dr. Ecolero at UNAM's Geology Institute. The proposed problem was to solved three objective functions: maximize the total water extraction, minimize the average drawdown and minimize the average drawdown velocity The results shows that the system converges to four different types of optimal solutions: one that favors the extraction, one that favors conservation and other two corresponding intermediate cases, where the extraction and conservation are balanced. These solutions, being part of a Pareto front, are completely equivalent from the viewpoint of multi-objective optimization, so we put them together in to a postoptimization analysis using Game Theory. we built a widespread game between three players, the management authority of the aquifer, the users of the resouce and a section of society whose main interest is to protect the aquifer. Calculating the Nash equilibrium associated, we found that the type of solution to be adopted is one of the two in which extraction and conservation are balanced. This shows that the proposed methodology is not only capable of generating optimal solutions, but also constitutes a valuable tool in the process of decision making and conflict resolution. Likewise, outline a future application in land management, which seeks to evaluate what would be the best place to put a new municipal dump in Tuxpan, Michoacán, considering morphope-dological properties, costs and environmental impacts. It is shown that when partial results obtained are in good agreement with independent field studies.

Capítulo 1

Introducción

§1.1. Estructura de la Tesis.

En el capítulo de introducción, se hace una justificación de la importancia del presente estudio, en el marco del uso sostenible de los recursos naturales.

En el capítulo 1 se desarrolla la teoría matemática sobre la cual se fundamenta este trabajo, el control óptimo. Se parte del control clásico, avanzando poco a poco hasta concluir con la formulación variacional del problema del control, mostrando como se puede aplicar en un problema de control de incendios forestales. Este es un capítulo muy teórico pero sustenta el resto del trabajo.

En el capítulo 2 se formula el problema de control óptimo del flujo subterráneo, presentando posteriormente los paradigmas computacionales en los cuales se realizó la implementación del Sistema de Manejo Óptimo de Recursos Naturales (SMORN), el computo evolutivo y la programación dinámica.

En el capítulo 3 se presenta la estructura general de SMORN, se introduce el sitio de estudio y se presentan los resultados obtenidos de la modelación matemática y computacional.

En el capítulo 4 se presentan tanto la discusión de los resultados como las conclusiones generales del trabajo de investigación. Adicionalmente se comenta sobre posibles aplicaciones futuras, así como su análisis desde la perspectiva de toma de decisiones.

Finalmente, en el apéndice se podrá encontrar una línea del tiempo sobre el desarrollo histórico del control óptimo, la licencia con la cual se edita el presente trabajo y una copia de algunos de los códigos usados en SMORN.

§1.2. Agua y desarrollo sustentable.

En los últimos años se ha subrayado en diferentes foros que tanto la disponibilidad como la calidad del agua son de los problemas más urgentes que afronta la humanidad. Al respecto, en el informe sobre el medio ambiente de la ONU (GEO 2000) se afirmó que la escasez mundial de agua constituye un problema de enorme trascendencia ya que “parece improbable que el ciclo hidrológico global, pueda adaptarse a las demandas que se presentarán en las próximas décadas”. Se reconoce además que el agua es parte integral del desarrollo sustentable y que de forma indirecta, incide sobre, contando ya al agua, las restantes áreas temáticas definidas en la Cumbre Mundial sobre el Desarrollo Sostenible (CMDS): energía, salud, agricultura, y biodiversidad.

No obstante a que en las Metas del Milenio sólo un punto hace referencia directa al recurso hídrico, un mejor manejo sobre él, puede tener un profundo impacto sobre el conjunto completo de las Metas del milenio como se muestra en la siguiente Cuadro 1.1.

§1.2.1. Recursos hídricos en crisis.

Los recursos hídricos, aunque considerados como renovables, solo lo son desde un punto de vista de su aprovechamiento antrópico, dentro de ciertos límites debido a su gran variabilidad tanto espacial como temporal en su disposición. Este elemento, la disponibilidad, constituye uno de los ejes centrales de las nuevas visiones de la gestión del recurso. De esta forma, una de las principales líneas de investigación en gestión óptima es superar la variabilidad y reducir las dificultades de predecir los flujos de los recursos hídricos, para conseguirlo que se conoce como aseguramiento del agua.

El problema de la escasez del agua, no es que el recurso se este agotando, pues debido al ciclo hidrológico global el volumen total de agua en el planeta es, en escalas de tiempo humanas, constante. El problema se trata entonces de en que compartimiento del ciclo hidrológico va a estar contenida la mayor cantidad de agua, es decir cuanta agua esta realmente disponible para nuestro uso. Se estima que actualmente , más de dos millones de personas se ven afectadas por la escasez de agua en más de 40 países: 1100 millones no tienen suficiente agua potable y 2400 millones no disponen de servicios de saneamiento [1]. Esto puede dar lugar a varios problemas sociales como:

- aumento de enfermedades, por ejemplo colera
- menor seguridad alimentaria
- conflictos entre usuarios

Concepto	Meta del milenio (2015)	Contribuciones directas	Contribuciones indirectas
Pobreza	Reducir a la mitad la proporción de personas cuyos ingresos son menores a 1 USD diario	<p>El agua es una pieza clave en la producción agrícola, industrial y en otras actividades económicas.</p> <p>La inversión en infraestructura y servicios relacionados al agua, actúan como catalizador para el desarrollo local y regional.</p>	<p>La disminución en la vulnerabilidad respecto de los peligros relacionados al agua, reducen el riesgo de inversión.</p> <p>Un manejo adecuado del recurso disminuye el impacto ambiental de su explotación, impulsando e desarrollo sustentable.</p> <p>La mejora de la salud, a través del aseguramiento de agua de buena calidad, aumenta la productividad.</p>
Hambre	Reducir a la mitad la proporción de personas que pasan hambre	<p>El aseguramiento del agua, permite mayor producción de grano.</p> <p>Agua de buena calidad es esencial en la agricultura de subsistencia, jardines domésticos, para ganado y frutales.</p> <p>Contribuye en la producción sostenible de pescado, frutas y otros alimentos de explotación comunitaria</p>	<p>Un manejo adecuado asegura la integridad de los ecosistemas, para mantener las corrientes de agua necesarias para la producción de alimentos.</p> <p>Partiendo de agua de mejor calidad, se puede abaratar la el costo de grano básico.</p>
Educación primaria universal	Asegurar que todos los niños completen la educación primaria		Mayor asistencia a la escuela debido a una mejor salud relacionada con la calidad del agua, así como a la reducción de tiempo y esfuerzo que supone transportar agua ara el consumo diario de las familias.
Igualdad entre los sexos	Que niños y niñas tengan igual acceso a la educación primaria y secundaria.		<p>Mayor asistencia de niñas, debido a que generalmente recae en ellas la responsabilidad de abastecer de agua a la familia.</p> <p>Las organizaciones comunales ara la gestión del agua mejoran el capital social de las mujeres.</p>
Mortalidad infantil	Reducir a dos tercios la tasa de mortalidad en niños menores de 5 años.	La mejor calidad y mayor cantidad de agua domestica y de saneamiento, reducen el primer factor de mortalidad en niños pequeños.	Ayuda al aseguramiento de alimentos y por tanto mejora la nutrición y salud, reduciendo así la propensión a enfermedades
Mortalidad materna	Reducir en tres cuartas partes la tasa de mortalidad materna	La mejora en salud y la disminución de cargas laborales por transporte de agua, reducen los riesgos de mortalidad	Las mejoras en salud y nutrición, reducen la propensión a la anemia y otras enfermedades que afectan la mortalidad materna.
Grandes enfermedades	Reducir a la mitad, detener y empezar a disminuir la difusión del VIH/SIDA, la malaria y otras grandes enfermedades	<p>Una mejor gestión del agua reduce los hábitats para mosquitos que transmiten la malaria.</p> <p>Menor incidencia de varias enfermedades, uno de cuyos vectores es la ingestión de agua deficiente.</p>	
Sustentable	Detener la explotación insostenible de recursos naturales y reducir a la mitad la proporción de gente que no tiene acceso a agua potable de buena calidad.	Una mejor gestión del agua, incluyendo control de contaminantes y volúmenes de extracción, son factores clave para mantener la integridad de los ecosistemas.	El desarrollo de una gestión integrada por cuencas fluviales crea las condiciones para que sea posible la gestión sostenible de los ecosistemas y se mitiguen los impactos entre cuencas y subcuencas.

Cuadro 1.1: Formas en como la gestión óptima del agua, puede coadyuvar a la obtención de las Metas del Milenio, establecidas por Naciones Unidas. Adaptada de Soussan (2002).

Lugar	Volumen ($10^3 Km^3$)	% Volumen total de la hidrosfera	Volumen reciclado anualmente (Km^3)	Periodo de renovación (años)	% Agua dulce
Océano	1338000	96.5	505000	2500	–
aguas subterráneas	234001	1.7	16700	1400	30.1
Humedad del suelo	16.5	.001	16500	1	.05
Glaciares y nieves perpetuas	24064	1.74	–	–	68.7
Lagos	76.4	0.013	10375	17	0.26
Marismas y pantanos	11.5	0.0008	2294	5	0.03
Ríos	2.12	0.0002	43000	16 días	0.006
Biológica	1.12	0.0001	–	–	0.003
Atmósfera	12.9	0.001	600000	8 días	0.04
Hidrosfera	1386000	100	–	–	–

Cuadro 1.2: Distribución del agua en el mundo, adaptado de [Shiklamanov, en preparación]

- limitaciones en diferentes medios de producción.

El Cuadro 1.2 muestra los volúmenes aproximados para el planeta en las diferentes fases del ciclo hidrológico, los volúmenes medios reciclados y sus tiempos medios de recuperación. Esta tabla resalta la enorme disparidad entre el inmenso volumen de agua salada y la pequeña fracción de dulce; así como el largo periodo de permanencia del hielo polar y las aguas subterráneas, comparado con la breve permanencia del agua en la atmósfera. Se estima así que aproximadamente 30 % del agua dulce del mundo está almacenada en los acuíferos subterráneos.

Algunas predicciones [39] muestran que para 2050 al menos una cuarta parte de la población vivirá en países afectados por la escasez crónica o recurrente de agua potable. En la actualidad esta tendencia puede apreciarse en la Fig1.1 donde se muestra la dificultad que encuentran muchos países para proveer los 1700 metros cúbicos de agua potable, que cada persona requiere para tener una vida activa y saludable.

Por otro lado, las corrientes de agua también son esenciales para la viabilidad de todos los ecosistemas. Por tanto, los niveles insostenibles de extracción de agua para otros usos disminuyen el total disponible para mantener la integridad de los ecosistemas. Debido al aumento poblacional y a que en 2020 se prevé que el 60 % de la población mundial vivirá en ciudades, donde se consume mayor cantidad de agua, es fácil concluir que nuestra actual forma de utilización del recurso conducirá inevitablemente a una mayor perturbación y degradación

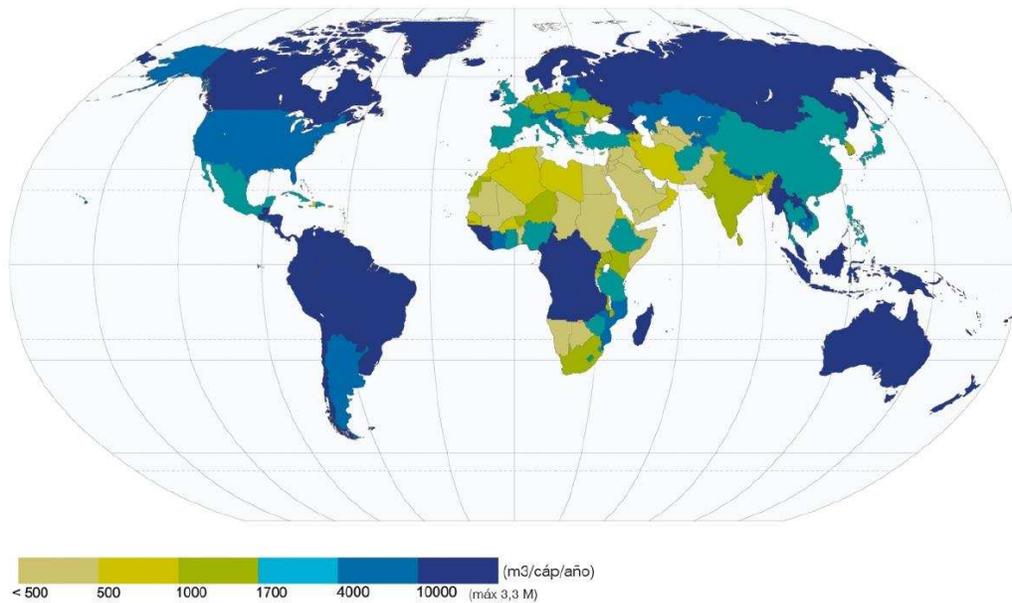


Figura 1.1: Disponibilidad de volumen total de agua interna renovable per capita. Mapa generado por el Programa Mundial de Evaluación del Agua (PMEA)

de los sistemas naturales.

Así, a medida que la población de los centros urbanos crece, también lo hacen sus demandas de recursos, como se muestra en la Fig.1.2, reflejando tanto las elevadas densidades poblacionales como los estilos de vida urbanos. Entre otras consecuencias, esto ha fomentado una sobrecarga en la infraestructura de abastecimiento de agua y saneamiento, forzando a muchas ciudades del mundo a recurrir con cada vez mayor frecuencia del agua subterránea. La situación se agrava aún más, al considerar que esos sistemas naturales juegan un papel fundamental en el control del ciclo hidrológico, de tal forma que al perturbarlos podríamos profundizar el problema de la disponibilidad del agua.

Esta situación se agrava por que alrededor de 263 cuencas fluviales son compartidas por dos o mas países, en los cuales incidentalmente vive el 40 % de la población del mundo y en los cuales los acuerdos internacionales necesarios para regular la equidad en el uso del recurso son débiles o inexistentes.

Se considera [24] que en México la crisis del agua no es una amenaza futura, sino un proceso que ya se hace sentir, como se puede apreciar en las Fig.1.3 donde se muestra la presión sobre el agua según las estadísticas del 2003, así como en la Fig.1.4 que muestra los acuíferos sobreexplotados en el país.

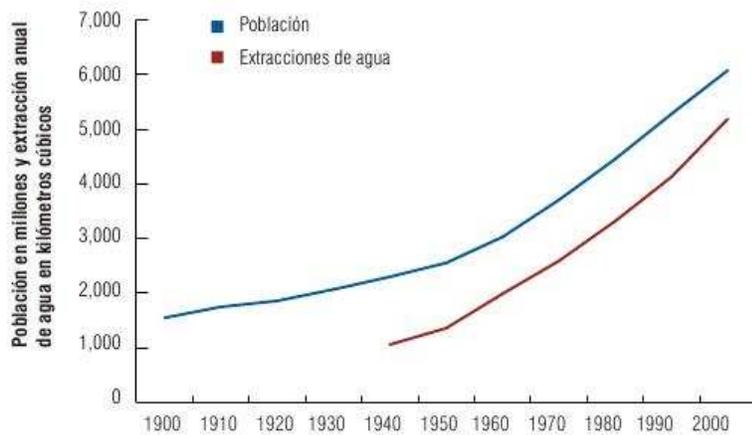


Figura 1.2: Correlación entre el crecimiento poblacional y extracción anual de agua dulce. Tomado de [gardner y engelman, 1997]

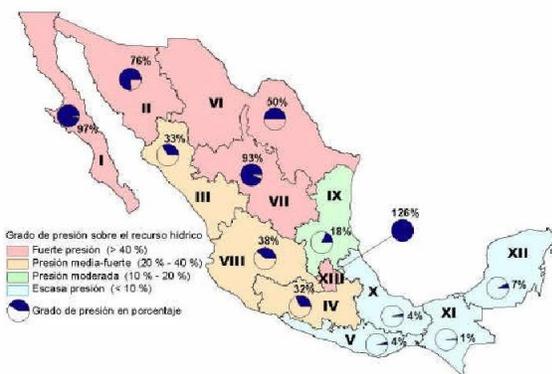


Figura 1.3: Presión sobre el recurso hídrico. Tomado de Carabias y Landa (2005)



Figura 1.4: Acuíferos sobreexplotados de México. Tomado de CONAGUA (2007)

Nombre del acuífero	Superficie ($10^6 Km^2$)	Valúen ($10^6 m^3$)	Tiempo rellenado	Continente
Areniscas de Nubia	2	75000	75000	África
del Norte del Sahara	0.78	60000	70000	África
de las Altas Planicies	0.45	15000	2000	Norte América
Guarani	1.2	30000	3000	Sur América
de la Llanura Norte de China	0.14	5000	300	Asia
Cuenca Artesiana	1.7	20000	20000	Australia

Cuadro 1.3: Lista de los mayores acuíferos del mundo [Margat, 1990a,1990b].

§1.2.2. Importancia de estudiar el agua subterránea

Del Cuadro 1.2 se puede entender la importancia del agua subterránea como reservorio de la tercera parte del agua dulce disponible en el mundo, razón por la cual desde los más remotos tiempos, los asentamientos humanos han obtenido buena parte de sus necesidades básicas de agua a partir de fuentes subterráneas, ya sean manantiales, norias, pozos, etc. Hoy con el desarrollo de la geología, la ingeniería civil, la tecnología de perforación y extracción; hemos alcanzado una tasa de extracción mundial de agua subterránea alrededor de $700 Km^3/año$. [77].

Así es como hoy día, el agua subterránea es el recurso natural más extraído en el mundo y constituye, por ejemplo, la piedra angular de la “revolución verde” de Asia, proporciona cerca del 70 % del agua potable en la Unión Europea y sustenta casi por si misma la vida rural de amplias áreas rurales del África subsahariana. En este mismo sentido se estima que alrededor del 50 % del abastecimiento actual de agua potable del mundo proviene del subsuelo, así como el 40 % del agua usada en la industria y el 20 % de la usada en agricultura. De hecho la lista de ciudades importantes y medianas que dependen considerablemente del agua subterránea es tan grande que basta decir que se estima que más de 1200 millones de habitantes en ciudades de todo el mundo dependen de pozos o manantiales [21, 22].

Así, la noción de gestión integral del agua en general y del agua subterránea en particular, ha fomentado un cambio progresivo de nuestro enfoque del desarrollo, hacia un planteamiento que tenga en cuenta, las complejas conexiones entre la sociedad, la cultura, la ciencia y el medio ambiente. En 1977, durante la Conferencia de Mar del Plata, se subrayó la importancia de considerar un enfoque interdisciplinario que incluyera la participación de ingenieros, economistas y especialistas del medio ambiente, entre otros.

Pero, sin una base sólida de investigación ¿cómo puede un país abordar adecuadamente los complejos problemas de la gestión del agua, tanto desde la perspectiva local como mundial? Para intentar dar una respuesta a esta pregunta, analicemos la Fig. §1.2.2 en la que se muestra

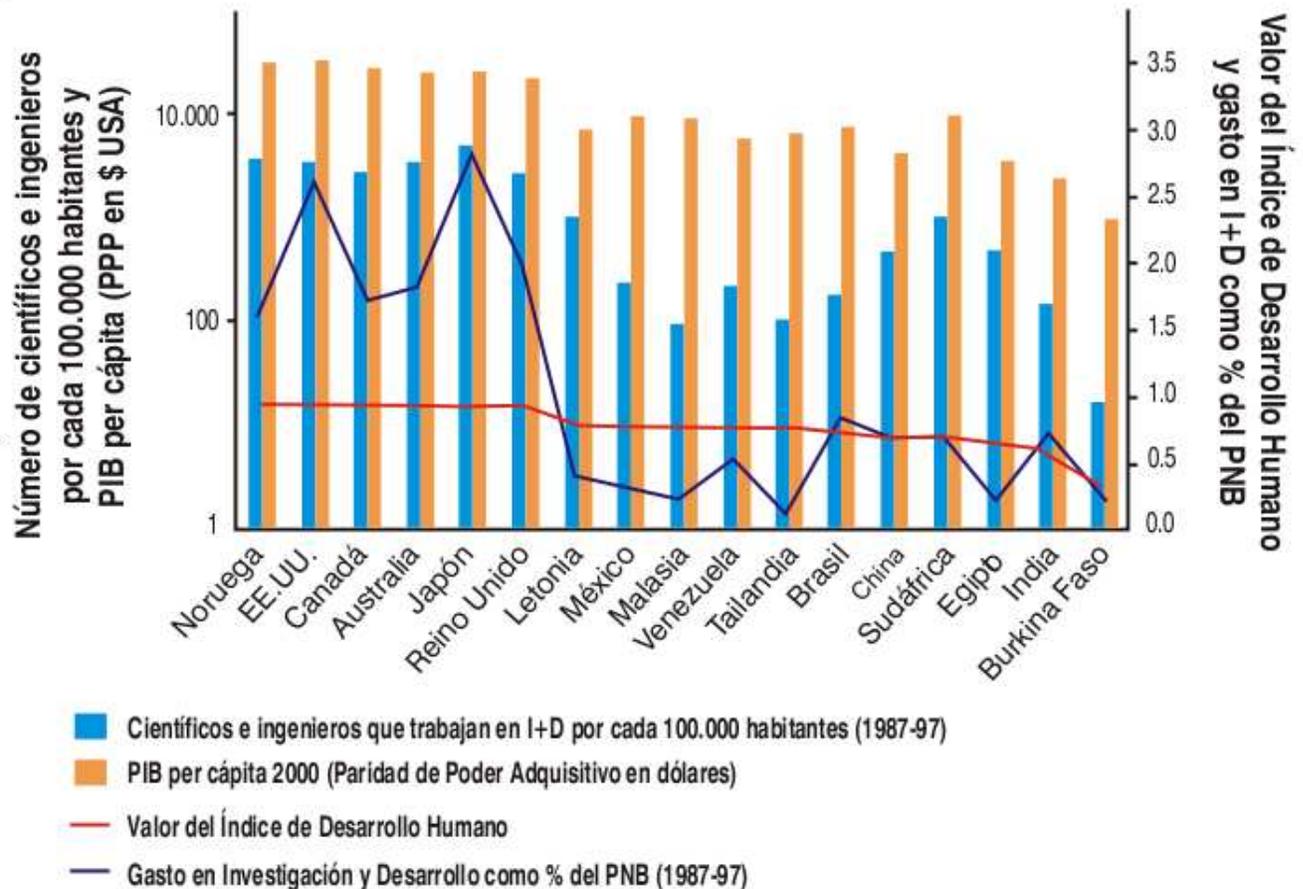


Figura 1.5: Número de científicos por cada 10000 habitantes, así como algunos indicadores de desarrollo económico, mostrando las diferencias entre países desarrollados y países en desarrollo.

el número de científicos e ingenieros por cada 100 000 habitantes, así como otros indicadores de desarrollo económico.

Por otro lado, en la Fig. §1.2.2 se muestra el porcentaje de dependencia de agua dulce por país. No nos sorprende que justamente los países que tienen el recurso asegurado son aquellos que en la Fig. §1.2.2 muestran mayor actividad científica y que además sean ellos quienes tengan mejores indicadores de desarrollo, es decir que por lo menos de forma heurística se puede reconocer una correlación entre ciencia, aseguramiento del agua y desarrollo.

Queda claro entonces que se debe generar conocimiento de los procesos básicos del agua, al tiempo que desarrollamos técnicas más eficientes de extracción y uso. Sin embargo, a pesar de que en estos países se han hecho avances significativos en cuanto a ciencia básica y tecnología; la vigilancia y gestión de los recursos hídricos, no ocupan un lugar importante en la agenda de investigación. Este aspecto negativo de las tendencias de investigación, se ve agravado por el hecho de que muchos obstáculos en el suministro eficaz del agua se encuentran en la

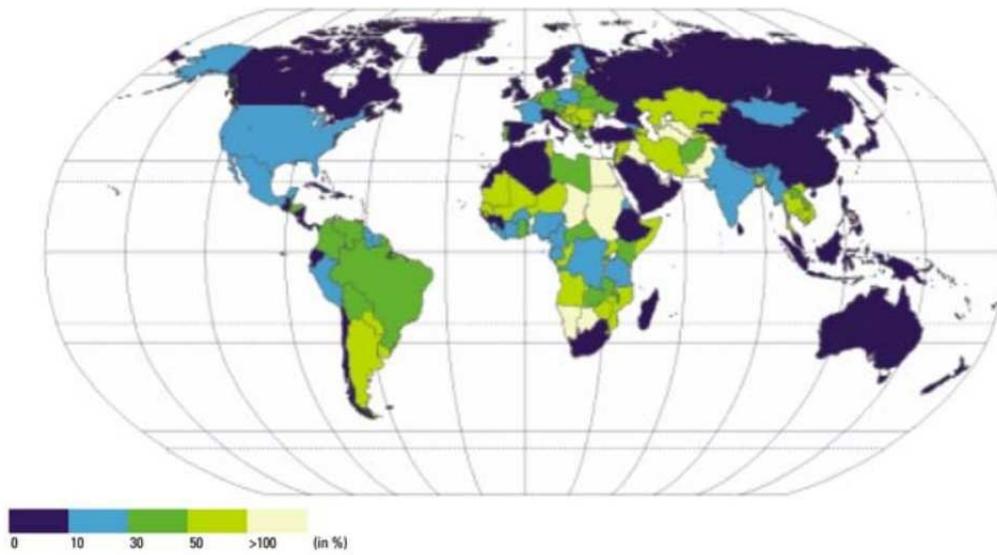


Figura 1.6: Porcentaje de dependencia externa de agua.

esfera de las instituciones y de la gestión, por lo cual urge fomentar proyectos de investigación centrados justamente en estructuras institucionales y técnicas de gestión óptima.

Capítulo 2

Teoría matemática de control óptimo

§2.1. Semblanza Histórica

La teoría de control es una rama interdisciplinaria de la ingeniería y las matemáticas, que estudia el comportamiento de distintos sistemas dinámicos, así como la forma de obtener una respuesta particular de ellos.

Las ideas detrás de la teoría de control podrían remontarse a los tiempos en los que el Hombre comenzó a crear tecnología. Ya en el diseño romano de acueductos se usaba un sistema de válvulas para mantener el nivel de agua constante e incluso se ha sugerido que este tipo de control incipiente en los sistemas de irrigación era un arte bien dominado en Mesopotámia, hace unos 2000 años A. C. En el antiguo Egipto había trabajadores llamados “harpenodaptai” o “estiradores de cuerdas” que estaban especializados en estirar cuerdas para producir largos segmentos rectos. Evidencia de que incluso entonces se había comprendido uno de los problemas más clásicos de la teoría de control y del cálculo de variaciones; que la distancia más corta entre dos puntos es la línea recta. Pero por supuesto en aquellos días no existía una teoría capaz de englobar estos conocimientos, menos aún de explicarlos.

Durante mucho tiempo, la teoría de control estuvo dominada por la aproximación ingenieril, ejemplificada por el pensamiento de Aristóteles, quien en el capítulo 3 del primer volumen de su *Política* dice: “... si cada instrumento pudiera llevar a cabo su propia función, respondiendo o anticipándose al trabajo de otros ... si la lanzadera tejiese y la púa tocase el arpa sin una mano que los guiara, los patronos no necesitarían sirvientes ni capataces.”[11] Es decir, la visión de la Teoría de control como un medio para alcanzar la automatización de los procesos de producción, logrando así la liberación y mejora de la calidad de vida del ser humano.

De esta forma, el desarrollo de esta disciplina vio un crecimiento sorprendente durante los años de la revolución industrial, siendo un ejemplo claro el trabajo de Maxwell sobre

el controlador de velocidad para la maquina de vapor de Watt. Este trabajo, basado en lo que ahora conocemos control proporcional, [55] permitía regular la cantidad de combustible admitida por un motor, de tal forma que se pudiera mantener una velocidad casi constante. Un compañero de clase de Maxwell, Edward John Routh, generalizó este resultado para toda la clase de sistemas lineales [65]. De forma independiente Adolf Hurwitz llegó al mismo resultado pero analizando la estabilidad del sistema usando ecuaciones diferenciales, llegando así a lo que se conoce como teorema Routh-Hurwitz [66, 47]. Otro ejemplo de esta incipiente aplicación de la teoría de control, se encuentra en los trabajos de Huygens y Hooke sobre la oscilación del péndulo a finales del siglo XVII; donde se reconoce el interés de generar instrumentos que sirviesen a la navegación marítima.

No obstante, este desarrollo continuo, fue hasta la década de los 60's del siglo pasado que pudo formularse una estructura completa que pudiera reconocerse como una teoría consistente la cual es conocida como Teoría de Control Clásica (TCC) . Es también a partir de esta década que se reconoce la necesidad de adentrarse en el mundo de la dinámica No Lineal para poder explicar satisfactoriamente la complejidad del mundo real. En este sentido, las contribuciones de Bellman[9] (programación dinámica), Kalman [50](filtrado y análisis algebraico de problemas de control) en los Estados Unidos y de Pontryagin[63] (Principio del Máximo en el control óptimo no-lineal) en la Unión Soviética, establecieron las bases fundamentales de la investigación en teoría de control.

Hasta ahora hemos mencionado fundamentalmente los avances logrados desde la ingeniería, sin embargo desde sus inicios pero sobre todo en épocas recientes la teoría de control ha progresado desde las matemáticas aplicadas, en particular del cálculo de variaciones. Esta disciplina podría considerarse que surgió con las observaciones de Herón de Alejandría en su obra “La Catoptrique”, en la cual mostró que la ley de reflexión de la luz puede deducirse de lo que ahora reconoceríamos como un principio variacional simple, que garantiza que la luz sigue el camino más corto. En el siglo XVII Fermat generalizó este principio y formuló el “principio del tiempo mínimo” según el cual la luz sigue el camino que garantiza el tiempo mínimo de recorrido. Posteriormente Leibniz y Huygens mostraron que la ley de Snell de la refracción se puede obtener también a partir del principio de Fermat. En 1696 J. Bernoulli resolvió otro famoso problema, el de la curva braquistocrona, que busca encontrar una curva en el plano cartesiano que vaya del punto $P = (x_0, y_0)$ al origen de modo que un punto material que se desliza sin fricción sobre ella tarde el menor tiempo posible en ir de P al origen. Más notas históricas sobre el desarrollo del cálculo de variaciones se pueden consultar en [69]. Es tanta la influencia matemática que existe en la actual teoría de control, que uno de sus protagonistas R. Kalman señaló en un artículo de 1974, que los avances en teoría de control y optimización de sistemas complejos, vendrían de los avances de la matemática más que de

los tecnológicos. En la actualidad, hay un gran interés por entender los sistemas de control de dimensión infinitas, formulados en términos de Ecuaciones en Derivadas Parciales (EDP) [78].

Para ver una línea de tiempo parcial sobre el desarrollo de la teoría del control, revisar el apéndice A1.

§2.2. Conceptos fundamentales.

La palabra *control* tiene dos acepciones comunes. La primera esta relacionada con el proceso de probar si un dispositivo físico o matemático tiene un comportamiento satisfactorio. La segunda se relaciona con el acto de controlar, de llevar a cabo acciones que garanticen que un sistema físico se comporte de un modo particular. Más adelante se dará una definición formal en términos matemáticos.

Como vimos en el párrafo anterior, el control es un conjunto de acciones que se ejercen sobre un *sistema*, el cual puede ser definido como un conjunto de componentes que interactúan, trabajando en conjunto para cumplir un objetivo determinado.

A pesar de que la formulación matemática de la teoría de control es intrínsecamente compleja, las ideas fundamentales que la soportan son suficientemente simples e intuitivas.

El primer concepto fundamental es el de la *retroalimentación*, que se entiende como el proceso por el cual una variable de salida del sistema que esta siendo controlado, determina la forma en como actuará el control en el siguiente paso de tiempo. Aunque poco reconocido de esta forma, una de las aportaciones más importantes de Darwin [27] es el haber reconocido que la selección natural es un proceso de retroalimentación en sistemas biológicos, que actúa sobre grandes periodos de tiempo. De igual forma Volterra [48] ocupó este concepto para explicar la dinámica poblacional entre dos poblaciones de peces. Sin embargo, quien formuló el concepto de retroalimentación como actualmente se usa dentro de la teoría de control fue Wiener (1949) y comenzó a ser usado de forma generalizada en la década de los 70's del por los ingenieros de los laboratorios Bell [75, 56].

El segundo concepto básico es el de las *fluctuaciones*, cuya idea fundamental es que muchas veces no necesitamos llevar de golpe al sistema bajo control al estado deseado. En muchos casos resulta ser más eficiente, y físicamente realizable, controlar al sistema pero dejarlo fluctuar, buscando el control que nos permita alcanzar el estado deseado sin forzar demasiado al sistema.

La necesidad de mantener cierto nivel de fluctuaciones es un principio general que se encuentra también en otras áreas de las matemáticas [3, 4, 6, 7], así como en la teoría política:

“Es un hecho curioso que, mientras que los economistas políticos reconocen que para un correcto funcionamiento de la ley de la oferta y la demanda ha de haber fluctuaciones, esto no ha sido reconocido por los mecánicos en relación a la máquina de vapor. El objetivo de los economistas no es suprimir estas fluctuaciones completamente (puesto que entonces se suprimir también el principio de la auto regulación), sino disminuirlas lo más posible, permitiéndoles que sean lo bastante grandes como para mantener suficiente poder regulador” [41].

El tercer concepto y entorno al cual gira el grueso de esta tesis es la idea de la *optimización*. Esta es una rama muy bien establecida de las matemáticas, cuyo objetivo es encontrar los valores adecuados de los parámetros que caracterizan a un sistema de tal forma que se maximiza o minimiza alguna cantidad de interés (como el beneficio o costo). La relación de la optimización con la teoría de control estriba en que para muchos sistemas físicos, alcanzar un estado particular de forma exacta es prácticamente imposible, por lo que el problema de control se puede formular en términos de encontrar los controles adecuados que lleven al sistema lo más cerca posible (minimizar la diferencia entre estado deseado y alcanzable) del estado deseado [5, 14]. Esta aproximación a la teoría de control ha sido muy fructífera porque los problemas de optimización suelen ser resueltos mediante sistemas de ecuaciones diferenciales, los que a su vez pueden plantearse de forma matricial. Esta posibilidad de trabajar con problemas matriciales nos permite utilizar todos los resultados del álgebra lineal, así como la programación dinámica [9] y el principio de Pontryagin [63].

§2.3. Teoría clásica de control.

Debido a que este trabajo se orienta hacia la formulación matricial de los problemas de control, en esta sección solo haremos un breve esbozo de la teoría clásica de control [60, 61]. Esta rama del control se basa en la obtención de las llamadas funciones de transferencia, que son ecuaciones en el espacio de las frecuencias que caracterizan las relaciones de entrada-salida de sistemas lineales invariantes en el tiempo.

La función de transferencia es el cociente de la transformada de Laplace de la salida (función de respuesta) y la transformada de Laplace de la entrada (función excitadora), bajo la suposición de que todas las condiciones iniciales son cero.

Entonces, sea un sistema lineal invariante en el tiempo definido por la siguiente ecuación diferencial:

$$a_0 y^{(n)} + a_1 y^{(n-1)} + \dots + a_{n-1} \dot{y} + a_n y = b_0 x^{(m)} + b_1 x^{(m-1)} + \dots + b_{m-1} \dot{x} + b_m x \quad (n \geq m), \quad (2.1)$$

donde y es la salida del sistema y x es la entrada. Entonces se consigue la función de transferencia tomando la transformada de Laplace.

$$\text{Función de Transferencia} \doteq \frac{Y(s)}{X(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m}{a_0 s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n}.$$

Es importante hacer notar que la función de transferencia es una propiedad del sistema, independiente de la forma de las funciones excitadoras y de respuesta. Sin embargo, no provee ninguna información respecto de la estructura física del sistema.

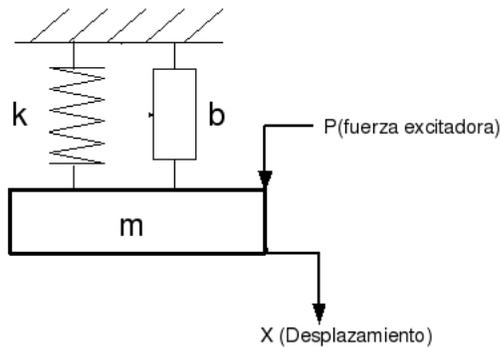


Figura 2.1: Sistema masa-resorte-amortiguador

Para clarificar este concepto veamos el ejemplo de un sistema masa(m)-resorte(k)-amortiguador(b) mostrado en la Fig. 2.1. Para obtener la función de transferencia asociada, se suelen seguir los siguientes pasos:

1. Escribir la ecuación diferencial del sistema, la ecuación gobernante.
2. Obtener la transformada de Laplace de la ecuación gobernante, suponiendo condiciones iniciales cero.
3. Obtener explícitamente la función de transferencia haciendo la relación entre $Y(s)$ y $X(s)$.

Aplicando la segunda ley de Newton al sistema dado, obtenemos la ecuación gobernante

$$m\ddot{x} + b\dot{x} + kx = p.$$

Tomando ahora la transformada de Laplace en ambos miembros de la ecuación obtenemos

$$\mathcal{L}\{m\ddot{x} + bx + kx\} = \mathcal{L}\{p\},$$

$$m[s^2X(s) - sx(0) - \dot{x}(0)] + b[sX(s) - x(0)] + kX(s) = P(s).$$

Haciendo nulas las condiciones iniciales, obtenemos entonces que

$$(ms^2 + bs + k)X(s) = P(s).$$

Con lo que la función de transferencia es

$$G(s) = \frac{P(s)}{X(s)} = \frac{1}{ms^2 + bs + k}. \quad (2.2)$$

Una virtud de la aproximación por funciones de transferencia es que éstas son independientes de la estructura física del sistema, por lo tanto surge la posibilidad de describir muchos sistemas físicos diferentes, mediante la misma función de transferencia.

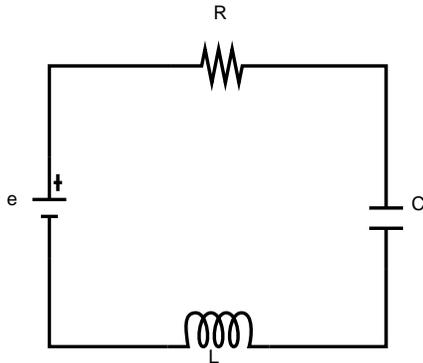


Figura 2.2: Circuito RLC

Tomando por ejemplo el circuito eléctrico de la Fig.2.2 en el cual e_i es el voltaje de entrada y e_o es el de salida. El circuito consta de una inductancia L, una resistencia R y un capacitor C. Aplicando las leyes de Kirchhoff el sistema está descrito por

$$\begin{cases} L \frac{di}{dt} + Ri + \frac{1}{C} \int i dt = e_i, \\ \frac{1}{C} \int i dt = e_o. \end{cases}$$

Al tomar la transformada de Laplace para calcular la función de transferencia, tenemos

$$\begin{cases} LsI(s) + RI(s) + \frac{1}{Cs}I(s) = E_i(s), \\ \frac{1}{Cs}I(s) = E_o(s). \end{cases}$$

Por tanto la función de transferencia es

$$G(s) = \frac{E_o(s)}{E_i(s)} = \frac{1}{LCs^2 + RCs + 1}, \tag{2.3}$$

que al compararla con §2.3, vemos que tienen la misma estructura, cambiando solo los parámetros físicos. Esto nos conduce a la conclusión de que debe ser válido hacer analogías entre diferentes tipos de sistemas físicos, haciendo las correspondientes analogías entre sus parámetros, para aplicaciones en ecología se puede consultar el trabajo de Peter Taylor [70].

Podemos de esta forma tener dos analogías mecano-eléctricas, la primera donde hacemos equivalentes la Fuerza-Voltaje (F-V); la segunda donde la equivalencia es entre Fuerza-Corriente (F-C), como mostramos en el Cuadro 2.1. La posibilidad de analizar un sistema mecánico usando la analogía eléctrica es sumamente práctica, pues la deducción de las ecuaciones gobernantes resulta ser más sencilla, sobre todo cuando el sistema mecánico involucra cuerpos rígidos, además de que hay una gama bastante amplia de simuladores de circuitos eléctricos lo cual hace inmediata su simulación computacional.

Sistema mecánico	Sistema Eléctrico (F-V)	Sistema Eléctrico (F-C)
Fuerza (par)	voltaje	Corriente
Masa (momento inercia)	Inductancia	Capacitancia
Coefficiente de fricción viscosa	Resistencia	Recíproco resistencia
Constante de resorte	Recíproco capacitancia	Recíproco inductancia
Desplazamiento	Carga	Flujo magnético
Velocidad	Corriente	voltaje

Cuadro 2.1: Analogías entre sistemas mecánicos y eléctricos.

Otra bondad de esta aproximación, es que una vez que se ha calculado la función de transferencia para un arreglo de dispositivos, su resultado es de nueva cuenta independiente del sistema físico que se analice. De esta forma, sistemas muy complicados pueden irse analizando poco a poco con ayuda de su correspondiente diagrama de bloques.

Un *diagrama de bloques* de un sistema es una representación gráfica de las funciones realizadas por cada componente, así como del correspondiente flujo de señales. Un bloque se



Figura 2.3: Elemento de un diagrama de bloque

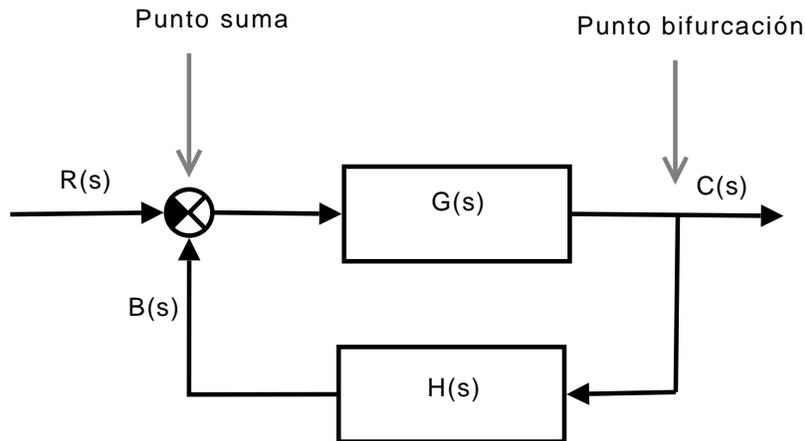


Figura 2.4: Diagrama general de un lazo cerrado

puede pensar como la operación matemática que el componente realiza sobre la señal que entra en él. Las funciones de transferencia de cada componente suelen ponerse dentro del bloque correspondiente, los cuales están unidos por flechas que indican el flujo de señales, como se muestra en la Fig.2.3.

La utilidad de trabajar con diagramas de bloques es que suele ser más fácil visualizar el funcionamiento de un sistema a partir de su diagrama que examinando el sistema físico en sí. Cabe aclarar que un diagrama de bloques contiene toda la información dinámica del sistema que representa, pero no hace ninguna referencia a su constitución física, así como tampoco indica explícitamente la fuente principal de energía. De esta forma para un mismo sistema podríamos tener varios diagramas de bloques equivalentes. Esto que inicialmente parecería una desventaja, termina funcionando a nuestro favor, pues sistemas muy complejos que tienen por lo común diagramas igualmente complicados, pueden irse reduciendo a diagramas cada vez más simple, siempre y cuando mantengamos la equivalencia en cada paso.

Pensemos en un diagrama de lazo cerrado como el de la Fig.2.4, donde la salida $C(s)$ es nuevamente alimentada a un punto suma donde se le compara con la entrada de referencia

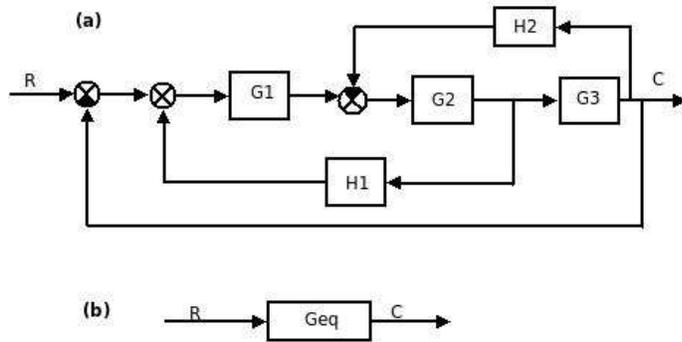


Figura 2.5: Reducción de un diagrama de bloques complejo

$R(s)$. En general, cualquier sistema físico se puede representar por un diagrama compuesto de bloques, puntos sumas y puntos de bifurcación¹. La relación de la señal de realimentación $B(s)$ con la señal de comparación, o error $E(s)$, se denomina función de transferencia de lazo abierto y esta dada por $\frac{B(s)}{E(s)} = G(s)H(s)$. Por su parte, la relación entre la señal de salida $C(s)$ con la de error $E(s)$ se denomina función de transferencia directa y esta dada por $\frac{C(s)}{E(s)} = G(s)$. Del punto suma tenemos que $E(s) = R(s) - B(s)$, que al combinarse con las ecuaciones anteriores nos da la función de transferencia del lazo cerrado completo

$$C(s) = \frac{G(s)}{1 + G(s)H(s)}R(s). \quad (2.4)$$

De esta forma, utilizando el resultado de (2.4), si tuviéramos un diagrama más complejo como el mostrado en la Fig.2.5 (a), podríamos ir reduciendo el diagrama para obtener el equivalente de la Fig. 2.5(b) cuya función de transferencia es

$$G_{eq} = \frac{G_1G_2G_3}{1 - G_1G_2H_1 + G_2G_3H_2 + G_1G_2G_3}.$$

Nótese que el numerador de esta función de transferencia de lazo cerrado es el producto de las funciones de transferencia en el paso de alimentación directa. El denominador es igual a $1 - \sum$ (producto de las funciones de transferencia al rededor de cada lazo).

Si en lugar de tener una sola señal de entrada y una sola de salida, tuviéramos un sistema con m entradas y n salidas como el de la Fig.2.6 , entonces tendríamos un vector de señales de entrada, uno de señales de salida y una matriz de funciones de transferencias. Lo cual nos daría una relación de transferencia

¹Un punto suma es un punto donde dos señales son sumadas o restadas, generalmente se usan para comparar señales. Un punto de bifurcación, es un punto a partir del cual la señal de salida de un bloque se dirige a otro bloque o punto suma.

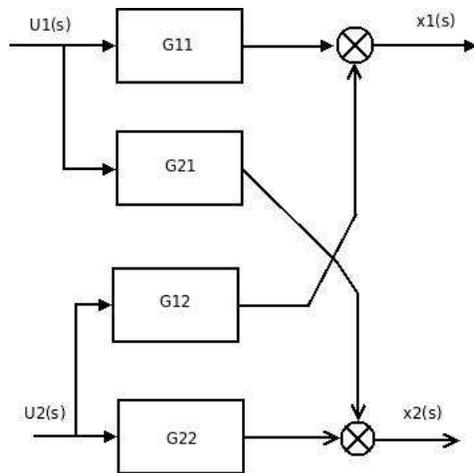


Figura 2.6: Sistema con múltiples entradas y múltiples salidas.

$$\begin{pmatrix} x_1(s) \\ x_2(s) \end{pmatrix} = \begin{pmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{pmatrix} \begin{pmatrix} U_1(s) \\ U_2(s) \end{pmatrix},$$

o de forma general

$$x_i(s) = G_{ij}(s)U_j(s); \quad i = 1, \dots, n; \quad j = 1, \dots, m.$$

Ahora bien, como seguramente se sospecha ya, conforme se vayan complicando los sistemas de estudio será cada vez más laboriosa la simplificación del diagrama de bloques, por lo cual suele dejarse de lado esta metodología para usar la teoría de gráficas.

Un gráfico de flujo de señal es un diagrama que representa un conjunto de ecuaciones algebraicas lineales, como las que se obtienen al aplicar la transformada de Laplace a las ecuaciones dinámicas que representan a un sistema.

Este tipo de gráficos consisten en redes cuyos nodos están conectados por ramas (líneas) que poseen dirección y sentido. Cada nodo representa una variable del sistema, mientras que las ramas actúan como multiplicadores de la señal. Por claridad, haremos algunas definiciones básicas en la teoría de gráficos:

- Nodos: se utilizan para expresar variables.
- Ramas: Son segmentos lineales que tienen ganancias y direcciones asociadas. La señal se transmite a través de una rama solamente en la dirección de la flecha.
- Nodo de entrada (fuente): Es un nodo que tiene solamente ramas de salida.

- Nodo de salida (pozo): Es un nodo que tiene solamente ramas de entrada.
- Trayectoria: es una sucesión continua de ramas que se dirigen en la misma dirección.
- Trayectoria directa: es una trayectoria que empieza en un nodo de entrada y termina en un nodo de salida, a lo largo de la cual ningún nodo se atraviesa más de una vez.
- Lazo: es una trayectoria que se origina y termina en el mismo nodo y en donde ningún otro nodo se atraviesa más de una vez.
- Ganancia de la trayectoria: es el producto de las ganancias de las ramas de una trayectoria. En el contexto del control, son las funciones de transferencia.
- Lazos disjuntos: Son lazos que no comparten ningún nodo en común.

La ventaja de usar la formulación por gráficos de flujo estriba en que en este contexto se cuenta con un resultado teórico, función de ganancia de Mason, la cual relaciona a las variables del sistema sin necesidad de realizar ninguna simplificación.

Formula de ganancias de Mason

La ganancia total de un gráfico de flujo esta dada por

$$P = \frac{1}{\Delta} \sum_k P_k \Delta_k,$$

donde P_k es la ganancia de trayecto del k-ésimo trayecto directo, Δ_k el cofactor del determinante del k-ésimo trayecto directo del gráfico, con los lazos adjunto del k-ésimo trayecto directo quitados y

$$\Delta = 1 - \sum_a L_a + \sum_{bc} L_b L_c - \sum_{d,e,f} L_d L_e L_f + \dots, \Delta_k$$

el determinante del gráfico.

Tomando como ejemplo el sistema representado por la Fig.2.5, quisiéramos determinar usando la formula de Mason, la función de transferencia de lazo cerrado $C(s)/R(s)$. Para esto, transformamos el diagrama de bloques en el gráfico de flujo

donde identificamos

$$\begin{aligned} L_1 &= G_1 G_2 H_1 \\ L_2 &= -G_2 G_3 H_2 \\ L_3 &= -G_1 G_2 G_3 \end{aligned}$$

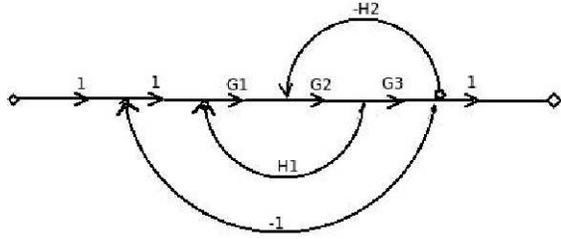


Figura 2.7: Gráfico de flujo para el diagrama de la Fig.2.5

con lo que

$$\begin{aligned}\Delta &= 1 - (L_1 + L_2 + L_3) \\ &= 1 - G_1G_2H_1 + G_2G_3H_2 + G_1G_2G_3.\end{aligned}$$

Luego, como $\Delta_1 = 1$, tenemos que

$$\begin{aligned}\frac{C(s)}{R(s)} &= \frac{P_1\Delta_1}{\Delta} \\ &= \frac{G_1G_2G_3}{1 - G_1G_2H_1 + G_2G_3H_2 + G_1G_2G_3}\end{aligned}$$

que es la misma función de transferencia que habíamos obtenido anteriormente.

Como mencionamos, la fórmula de Mason es particularmente útil cuando tenemos sistemas muy complejos a los cuales no es fácil calcularles la función de transferencia mediante la reducción de su diagrama de bloques.

§2.4. Teoría moderna de control.

Para la teoría clásica del control, que desarrollamos brevemente en la sección §2.3, el objetivo central es encontrar los mecanismos de control necesarios para llevar al sistema en estudio a un estado particular deseado. A este tipo de planteamiento se le conoce como *Controlabilidad*. En contra parte, la teoría de control óptimo, renuncia a obtener un estado particular exacto y en su lugar busca los controles necesarios para obtener el estado que más se aproxime al deseado. Esta aproximación llamada de *Optimización*, tiene la virtud de ser sumamente realista, con aplicabilidad directa y susceptible de ser abordada mediante la modelación computacional. Es alrededor de esta visión de optimización, que nos centraremos en el resto de este trabajo, por lo que en esta sección construiremos los resultados matemáticos formales que necesitaremos posteriormente [33, 78, 68].

Comenzamos por un ejemplo sencillo. Pensemos en un sistema cuyo estado esta definido por el vector

$$x_i = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}; x_i \in \mathbb{R}$$

y que esta gobernado por

$$A_{ij}x_j = b_i, \tag{2.5}$$

donde supondremos por simplicidad, para este ejemplo, que A_{ij} es una matriz cuadrada de $n \times n$, definida positiva, no singular e incluso simétrica. Por su parte el vector b_i que aparece en el lado derecho de la ecuación, es en si el control. De esta forma (2.5) describe la forma en como el control b_i actúa sobre el estado x_j . Nos imponemos entonces la condición

$$x_1 = \tilde{x}_1 \tag{2.6}$$

El problema de control óptimo asociado será entonces encontrar los $b_i \in \mathbb{R}$ tal que sean solución de (2.5) y satisfaga también (2.6) . Este problema es por supuesto trivial pues disponemos de más controles que componentes del estado que deseamos controlar, es claro que habrá entonces una redundancia de soluciones, la más evidente $b_i = A_{i1}\tilde{x}_1$. Sin embargo, veremos que el problema se complica si vamos reduciendo el margen de maniobra. Supongamos que restringimos b_i estén fijos para $i \neq n$, entonces, el nuevo sistema puede escribirse como

$$A_{ij}x_j = c_i + \tilde{b}_i, \tag{2.7}$$

donde $c_i \in \mathbb{R}$ es un vector fijo, mientras que $\tilde{b}_i = (0, \dots, b_n)$. Entonces, lo que nos preguntamos es si existe una b_n tal que x_j sea solución de (2.7) y satisfaga al mismo tiempo a (2.6).

La cuestión es ahora mucho menos obvia, pero podemos resolverla a través del siguiente cambio de variable

$$x_j = y_j + z_j,$$

donde $y_j = A_{jk}^{-1}c_k$, y z_j satisface $A_{ij}z_j = b_n$. Se puede mostrar entonces que el problema tiene solución cuando la primera componente de la inversa de A es no nula. Cuando esto no se satisface $z_1 = 0$, lo que implica que sea cual sea b_n , $x_1 = y_1$. De esta forma x_1 es insensible

al control o en términos formales, x_1 no es observable por el control b_i .

Ahora bien, dejando de lado el problema de controlabilidad y asumiendo una aproximación de optimización; la mejor respuesta al problema anterior consiste en maximizar el funcional

$$J(b_n) = |x_1 - \tilde{x}_1|^2,$$

en el intervalo $I_k = [-k, k]$.

Como J depende continuamente de b_n , se deduce inmediatamente la existencia de un control óptimo $b_n^k \in I_k$, que minimiza el funcional J .

De este ejemplo tan sencillo podemos hacer las siguientes observaciones:

- Si la condición de controlabilidad se cumple para una k suficientemente grande, la solución al problema de *optimización*, genera la solución exacta del problema de *controlabilidad*.
- Cuando \tilde{x}_1 no es *observable*, la solución al problema de *optimización* producirá de todas formas la mejor solución posible.
- Analizar el mínimo del funcional J conforme k crece, puede ser considerada en sí como una prueba de controlabilidad. Si éste mínimo se estabiliza alrededor de una constante positiva a medida que k aumenta, entonces podemos justificadamente sospechar que \tilde{x}_1 no es *observable*.

§2.4.1. Formulación del problema básico de control óptimo.

Definiremos la dinámica de un sistema como la ecuación diferencial ordinaria (EDO)

$$\begin{cases} \dot{x}_i(t) = f_i(x_i(t), \alpha(t)) \\ x_i(0) = x_i^0 \end{cases}, \quad (2.8)$$

y la trayectoria $x_i(\cdot)$ como la correspondiente respuesta del sistema.

Hay que hacer notar que nuestra solución $x_i(\cdot)$ de (2.8) depende tanto de $\alpha_i(\cdot)$ como de las condiciones iniciales (CI).

Notación:

Al escribir el funcional f_i consideramos

$$f_i(x, a) = \begin{pmatrix} f_1(x, a) \\ \vdots \\ f_n(x, a) \end{pmatrix}, \text{ y de manera similar } x_i(t) = \begin{pmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{pmatrix}.$$

Por otro lado, definimos al conjunto de controles admisibles como

$\mathcal{A} = \{\alpha_i : [0, \infty) \rightarrow \mathcal{A} | \alpha_i(\cdot) \text{ medible}\}$, donde

$$\alpha_i(t) = \begin{pmatrix} \alpha_1(t) \\ \vdots \\ \alpha_n(t) \end{pmatrix}$$

Para conseguir nuestro objetivo final, que es obtener la mejor solución de (2.8), necesitamos contar con un criterio que nos permita discriminar de entre todas las posibles soluciones la mejor. Un candidato inmediato de criterio sería un *funcional recompensa* que cuantifique alguna propiedad deseable del sistema. Este funcional puede definirse como

$$P[\alpha_1(\cdot)] = \int_0^T r(x_i(t), \alpha_i(t)) dt + g(x_i(T)), \quad (2.9)$$

donde $x_i(\cdot)$ resuelve (2.8) para el control $\alpha_i(\cdot)$; $r : \mathbb{R}^n \times \mathcal{A} \rightarrow \mathbb{R}$ es la *recompensa actual* y $g : \mathbb{R}^n \rightarrow \mathbb{R}$, la *recompensa terminal*; siendo T el tiempo terminal. De esta forma lo que buscamos es un control, llamado óptimo, $\tilde{\alpha}_i(\cdot)$ que maximiza la recompensa, es decir

$$P[\tilde{\alpha}_i(\cdot)] \geq P[\alpha_i(\cdot)],$$

para todo control $\alpha_i(\cdot) \in \mathcal{A}$.

Esta tarea nos presenta los siguientes cuestionamientos matemáticos:

- ¿Existe un control óptimo?
- ¿Como podemos caracterizar matemáticamente a ese control óptimo?
- ¿Cómo se construye dicho control óptimo?

§2.4.2. Ejemplos de control óptimo básico.

Comencemos por un problema cuyo funcional de recompensa sea lo suficientemente intuitivo. Supongamos que tenemos una fábrica en la cual deseamos implementar un control sobre la producción.

Sea $x(t)$ = cantidad de producción al tiempo $t \geq 0$. Proponemos un control $\alpha(t)$ que representa una parte de esta producción que se reinvierte, donde $0 \leq \alpha(t) \leq 1$. Definido así el control, el problema a resolver es

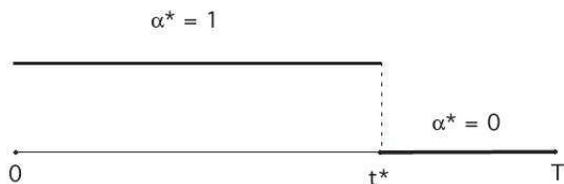


Figura 2.8: Control bang-bang, $\alpha^* \doteq \tilde{\alpha}$ en nuestra notación. Tomada de [33].

$$\begin{cases} \dot{x}(t) = k\alpha(t)x(t) \\ x(0) = x^0 \end{cases}, \quad (2.10)$$

con $k > 0$ un parámetro que modela la tasa de crecimiento de la reinversión. Para completar el problema, definimos un funcional de recompensa

$$P[\alpha(\cdot)] = \int_0^T (1 - \alpha(t))x(t)dt. \quad (2.11)$$

La interpretación de este problema es que dada la dinámica (2.10), queremos maximizar la cantidad de producción neta, habiendo descontado ya lo necesario para la reinversión.

Entonces identificamos $\mathcal{A} = [0, 1]$, $f(x, a) = k\alpha x$, $r(x, a) = (1 - \alpha)x$, $g = 0$ y como veremos más adelante, el control óptimo $\tilde{\alpha}(\cdot)$ será

$$\tilde{\alpha}(\cdot) = \begin{cases} 1; & 0 \leq t \leq \tilde{t} \\ 0 & \tilde{t} < t \leq T \end{cases}. \quad (2.12)$$

En otras palabras, para obtener la mayor producción neta debemos reinvertir todo hasta un tiempo crítico \tilde{t} y luego dejar de reinvertir. A este tipo de controles mostrados en la Fig.2.8 se les llama controles *bang-bang*.

Un segundo ejemplo que nos ilustra un método gráfico para resolver explícitamente un problema de control, es el problema del carrito propulsado por cohetes. Imaginemos que tenemos un carrito el cual es propulsado por cohetes colocados en sus extremos.

Definiendo las variables

$$q(t) = \text{posición del carrito al tiempo } t,$$

$$v(t) = \dot{q}(t) = \text{velocidad al tiempo } t,$$

$\alpha(t) = \text{propulsión por los cohetes};$

podemos describir la dinámica del sistema por

$$\begin{cases} \dot{x}(t) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \alpha(t) \\ x(0) = x^0 = (q^0, v^0)^T \end{cases} \quad (2.13)$$

y al funcional de recompensa como

$$P[\alpha(\cdot)] = - \int_0^T 1 dt = -T. \quad (2.14)$$

Siento T el primer tiempo para el cual $q(T) = v(T) = 0$.

Aunque lo justificaremos más adelante, pensemos que el control óptimo para este sistema es un control del tipo bang-bang, de tal forma que $\alpha = -1$ o $\alpha = 1$.

Supongamos que tenemos el caso donde $\alpha = 1$ para un cierto intervalo, en el cual

$$\begin{cases} \dot{q} = v \\ \dot{v} = 1 \end{cases}, \quad (2.15)$$

entonces $v\dot{v} = \dot{q} = \frac{1}{2}(v^2)$.

Si permitimos que t_0 este en el intervalo donde $\alpha = 1$ e integramos de t_0 a t :

$$\frac{v^2(t)}{2} - \frac{v^2(t_0)}{2} = q(t) - q(t_0)$$

y consecuentemente

$$v^2(t) = 2q(t) + \underbrace{(v^2(t_0) - 2q(t_0))}_b.$$

Dicho de otra forma, mientras el control sea $\alpha = 1$ las trayectorias del carrito en el espacio de configuración serán las curvas $v^2 = 2q + b$, con b una constante, como se muestra en la Fig.

Si consideramos ahora el caso en el que $\alpha = -1$, haciendo $\dot{v} = -1$ en (2.15), obtenemos que las trayectorias de la Fig.2.10, que ahora estarán dadas por $v^2 = -2q + c$, con c una constante.

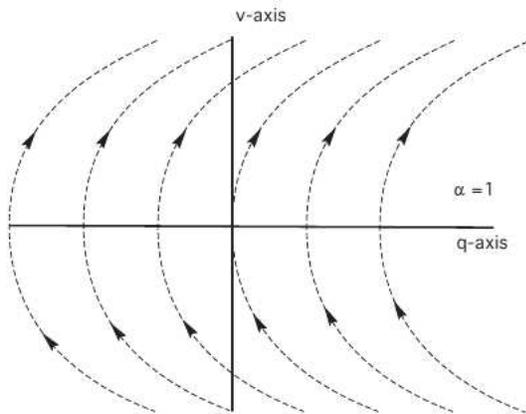


Figura 2.9: Trayectorias en el espacio de configuración cuando el control bang-bang esta en $\alpha = 1$. Tomada de [33].

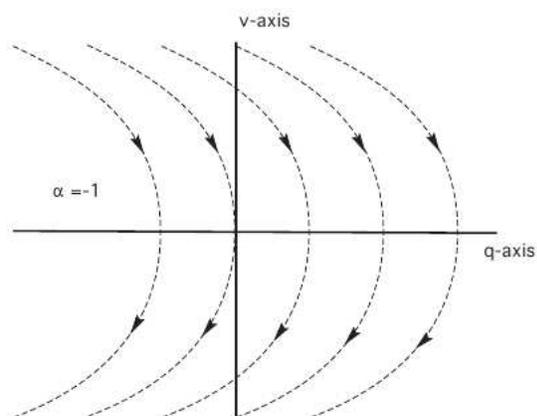


Figura 2.10: Trayectorias en el espacio de configuración cuando el control bang-bang esta en $\alpha = -1$. Tomada de [33].

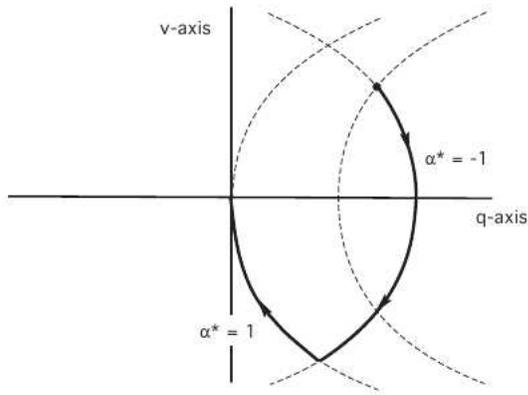


Figura 2.11: Trayectorias en el espacio de configuración para el control óptimo del problema del carro propulsado por cohetes. Tomada de [33].

Combinando ambos resultados como en la Fig.2.11, obtenemos que el control óptimo para el sistema es un control bang-bang con un solo cambio.

§2.4.3. Controlabilidad.

Ya en la subsección §2.4.1, presentamos al sistema (2.8)

$$\begin{cases} \dot{x}_i(t) = f_i(x_i(t), \alpha(t)) \\ x_i(0) = x_i^0 \end{cases},$$

donde $x_i^0 \in \mathbb{R}$, $f_i : \mathbb{R}^n \times \mathcal{A} \rightarrow \mathbb{R}^n$, $\alpha : (0, \infty] \rightarrow \mathcal{A}$ es el control y $x_i : (0, \infty] \rightarrow \mathbb{R}^n$ es la respuesta del sistema. Entonces, nos formulamos la pregunta: ¿dada una CI x^0 y un conjunto objetivo $S \in \mathbb{R}^n$, existe un control que lleve al sistema a S (en particular $S = 0$) en un tiempo finito? Estudiaremos ahora este cuestionamiento que fue ya esbozado en el primer ejemplo de §2.4, para lo cual dejaremos por el momento nuestro interés en el funcional de recompensa.

Definimos el *conjunto de estados alcanzables* $\mathcal{C}(t)$, como la colección de estados para los cuales existe un control tal que $x_i(t) = 0$. Y al *conjunto de estados alcanzables al tiempo t* $\mathcal{C} = \bigcup_{t \geq 0} \mathcal{C}(t)$, como la colección de estados para los cuales existe un control tal que $x_i(t) = 0$ en un tiempo finito t .

Suponemos además que (2.8) tiene una estructura lineal, de tal forma que puede escribirse como

$$\begin{cases} \dot{x}_i(t) = M_{ij}x_j(t) + N_{ik}\alpha_k(t) \\ x_i(0) = x_i^0 \end{cases} \quad (2.16)$$

De esta forma, lo primero que hay que averiguar es si el sistema tiene solución y que forma adoptara ésta. Para ello vale la pena hacer un breve recordatorio de la teoría de las EDO lineales del tipo (2.16).

Sea $X_{ij}(\cdot) : \mathbb{R} \rightarrow \mathbb{M}^{n \times n}$ definida mediante la matriz exponencial

$$X_{ij}(t) = e^{tM_{ij}},$$

tal que $x_i(t) = X_{ij}(t)x_j^0 = e^{M_{ij}t}x_j^0$ es solución única al sistema homogéneo

$$\begin{cases} \dot{x}_i(t) = M_{ij}x_j(t) \\ x_i(0) = x_i^0 \end{cases}.$$

Complementariamente, la solución al sistema inhomogeneo

$$\begin{cases} \dot{x}_i(t) = M_{ij}x_j(t) + f_i(t) \\ x_i(0) = x_i^0 \end{cases},$$

es

$$x_i(t) = X_{ij}(t)x_j^0 + X_{ik} \int_0^T X_{kl}(s)f_l(s)ds. \quad (2.17)$$

Aplicando la ecuación (2.17) para encontrar la solución de (2.16) con el control $\alpha(\cdot)$, obtenemos

$$x_i(t) = X_{ij}(t)x_j^0 + X_{ik} \int_0^T X_{kl}(s)N_{lm}\alpha_m(s)ds.$$

Se puede probar que $x_i^0 \in \mathcal{C}(t)$, es decir, el sistema es controlable, *si y solo si*

1.

Existe un control $\alpha(\cdot) \in \mathcal{A}$ tal que $x_i(t) = 0$,

2.

$$X_{ij}(t)x_j^0 + X_{ik} \int_0^T X_{kl}(s)N_{lm}\alpha_m(s)ds = 0 \text{ para algún } \alpha(\cdot) \in \mathcal{A} \text{ y}$$

3.

$$x_k^0 = - \int_0^T X_{kl}(s) N_{lm} \alpha_m(s) ds.$$

Basados en este resultado, podemos enunciar un criterio general de controlabilidad.

Criterio de Controlabilidad (CC).

Sea

$$G_{ij} = G(M, N)_{ij} = \underbrace{[N, MN, M^2N, \dots, M^{n-1}N]_{ij}}_{\text{matriz de } n \times (mn)} \quad (2.18)$$

entonces si $\text{rango}(G_{ij}) = n$, $\text{Re}(\lambda) < 0$ para todo valor propio de M_{ij} , entonces (2.16) es controlable.

La prueba de este teorema se basa en que $\text{rango}(G_{ij}) = n \iff 0 \in \mathcal{C}^{int}$ donde \mathcal{C}^{int} esta en el interior de \mathcal{C} . Pero quizás sea más didáctico mostrar la validez de este resultado, reanalizando el ejemplo del carrito propulsado por cohetes de la subsección §2.4.2.

En este ejemplo, para el cual $\mathcal{A} = [-1, 1]^m$; cuyo sistema es $x_i = M_{ij}x_j + N_{ik}\alpha_k$ con

$$n = 2, m = 1, \mathcal{A} = [-1, 1], M_{ij} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, N_{ij} = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

tendremos que

$$G_{ij} = [N, MN]_{ij} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

De esta forma $\text{rango}(G_{ij}) = 2$ y los valores propios λ están dados por

$$P(\lambda) = \det(\lambda I_{ij} - M_{ij}) = \det \begin{pmatrix} \lambda & -1 \\ 0 & \lambda \end{pmatrix} = \lambda^2,$$

por lo que al ser ambos valores propios cero, fallamos en satisfacer el teorema CC y por tanto el sistema no es controlable. Sin embargo hemos mostrado ya que a pesar de eso, se puede construir geoméricamente una solución en la formulación de optimización, por lo que pronto dejaremos la aproximación de control en beneficio de la optimización. Pero antes hablaremos de la variable dual² de la controlabilidad, la observabilidad.

Consideramos el sistema

²El termino dual aquí esta siendo usado en el contexto del álgebra lineal, donde por el Teorema Fundamental del Álgebra Lineal (TFAL) para un sistema lineal como 2.16, existen cuatro espacios vectoriales fundamentales: espacio renglón, espacio columna y sus correspondientes duales.

$$\begin{cases} \dot{x}_i(t) = M_{ij}x_j(t) \\ x_i(0) = x_i^0 \end{cases} \quad (2.19)$$

Supongamos que de este sistema podemos observar

$$y_i(t) = N_{ij}x_j(t), \quad (2.20)$$

como $M, N \in \mathbb{M}^{m \times n}$, una situación particularmente interesante surge cuando $m \ll n$. Podemos interpretar a $y(\cdot)$ como un conjunto de baja dimensionalidad de “observaciones” o “mediciones” de un sistema de alta dimensionalidad $x(\cdot)$.

Surge entonces un nuevo tipo de pregunta, dado un conjunto finito de observaciones $y(\cdot)$, ¿podemos reconstruir a $x(\cdot)$? ¿Provee $y(\cdot)$ suficiente información para deducir una condición inicial particular x^0 ?

Decimos entonces que el par (2.19),(2.20) es observable, si a partir de un $y(\cdot)$ para un intervalo cualquiera $[0, t]$, podemos obtener x^0 . Por supuesto tendremos los casos extremos

1. $N \equiv 0$, entonces claramente el sistema no es observable.
2. $m = n$, lo que implica que N es invertible, con lo cual $x_i(t) = N_{ij}^{-1}y_j(t)$ y por tanto el sistema es observable.

Por supuesto que los casos interesantes están entre éstos dos, par a los cuales se cuenta con el resultado general siguiente.

Teorema de Observabilidad (TO).

El sistema

$$\begin{cases} \dot{x}_i(t) = M_{ij}x_j(t) \\ y_i(t) = N_{ij}x_j(t) \end{cases} \quad (2.21)$$

será observable si y solo si el sistema

$$\dot{z}_i(t) = M_{ij}^T z_j(t) + N_{ik}^T \alpha_k(t), \quad \mathcal{A} \in \mathbb{R}^m \quad (2.22)$$

es controlable, es decir $\mathcal{C} \in \mathbb{R}^n$.

Para probarlo, supongamos que (2.21) *no es observable*, entonces $\exists x^1, x^2 \in \mathbb{R}$ tales que $x^1 \neq x^2$ y para los cuales

$$\begin{cases} \dot{x}_i^1(t) = M_{ij}x_j^1(t); x_i^1(0) = x^1 \\ \dot{x}_i^2(t) = M_{ij}x_j^2(t); x_i^2(0) = x^2 \end{cases},$$

pero $y_i(t) = N_{ij}x^1(t) = N_{ij}x^2(t)$ para todos los $t \geq 0$. Sea $x_i(t) = x_i^1(t) - x_i^2(t)$, $x^0 = x^1 - x^2$; entonces $\dot{x}_i(t) = M_{ij}x_j(t)$, $x_i(0) = x_i^0 \neq 0$, pero $N_{ij}x_j(t) = 0$ ($t \geq 0$).

Por otro lado, como $x_i(t) = X_i(t)x^0 = e^{tM_{ij}}x^0$, entonces

$$N_{ij}e^{M_{jk}t}x^0 = 0; (t \geq 0).$$

Derivando la expresión anterior l-veces respecto del tiempo, se puede demostrar que

$$N_{ij}M_{jk}^l x^0 = 0.$$

Entonces $(x^0)^T(M_{ij})^T(N_{jk})^T = 0$ y por tanto

$$[(N_{jk})^T, \dots, ((M_{ij})^T)^{n-1}(N_{jk})^T] = 0.$$

Como $x^0 \neq 0$, entonces el *rango* $[\cdot] < n$, con lo que se concluye que el sistema *no es controlable* y como partimos de la no observabilidad, tenemos entonces que no observable implica no controlable. Negando esta proposición lógica llegamos a que *observable implica controlable*.

Sin hacer todo el detalle, partiendo de que (2.21) es no controlable y usando el teorema de Cayley–Hamilton, se demuestra que no controlabilidad implica no observabilidad. Negando esta proposición y usando la primera parte de la prueba, se tiene finalmente que: *Un sistema es controlable si y solo si es observable*, que es el contenido del teorema (TO).

A pesar de que hemos avanzado un buen trecho en nuestro entendimiento de los problemas de control óptimo, una de las características más atractivas del mismo, es su relación con el cálculo variacional.

Sea una función suave $L : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$, $L = L(x, \dot{x})$; llamada lagrangiano; el problema básico del calculo variacional será entonces encontrar $\tilde{x}(\cdot) : [0, T] \rightarrow \mathbb{R}^n$, que minimiza un funcional llamado acción, definido como

$$S[x(\cdot)] = \int_0^T L(x(t), \dot{x}(t))dt, \tag{2.23}$$

de entre todas las posibles $x(\cdot)$ que satisfacen $x(0) = 0$, $x(T) = x^1$.

Supongamos que L tiene derivadas parciales continuas y extremiza la acción; entonces cualquier perturbación sobre x que preserve las condiciones de frontera, harán que la acción cerca (si se minimizaba) o decrezca (si se maximizaba).

Sea la perturbación $g(x) = x(t) + \epsilon\eta(t)$, donde η es una función diferenciable que satisface que $\eta(0) = \eta(T) = 0$. Entonces definimos a la acción perturbada

$$S_\epsilon = \int_0^T L(g(t), g'(t)) dt.$$

Si calculamos ahora la derivada total de la acción respecto de la perturbación, tenemos

$$\frac{dS_\epsilon}{d\epsilon} = \int_0^T \frac{dL(g(t), g'(t))}{d\epsilon} dt.$$

Por la definición de derivación total, se tiene

$$\frac{dL}{d\epsilon} = \frac{\partial L}{\partial t} \frac{\partial t}{\partial \epsilon} + \frac{\partial L}{\partial g} \frac{\partial g}{\partial \epsilon} + \frac{\partial L}{\partial g'} \frac{\partial g'}{\partial \epsilon} = \eta(t) \frac{\partial L}{\partial g} + \eta'(t) \frac{\partial L}{\partial g'},$$

de tal forma que

$$\frac{dS_\epsilon}{d\epsilon} = \int_0^T \left[\eta(t) \frac{\partial L}{\partial g} + \eta'(t) \frac{\partial L}{\partial g'} \right] dt.$$

De esta forma, cuando $\epsilon = 0 \Rightarrow g = x$ y como por construcción L es un extremo de la acción, entonces $S'_\epsilon(0) = 0$; es decir

$$\frac{dS_\epsilon(0)}{d\epsilon} = \int_0^T \left[\eta(t) \frac{\partial L}{\partial x} + \eta'(t) \frac{\partial L}{\partial x'} \right] dt = 0.$$

Integrando esta expresión por partes, tenemos

$$\frac{dS_\epsilon(0)}{d\epsilon} = \int_0^T \left[\frac{\partial L}{\partial x} - \frac{d}{dt} \left(\frac{\partial L}{\partial x'} \right) \right] \eta(t) dt + \left[\eta(t) \frac{\partial L}{\partial x'} \right]_0^T = 0.$$

Usando las condiciones de frontera en η se obtiene que

$$\frac{dS_\epsilon(0)}{d\epsilon} = \int_0^T \left[\frac{\partial L}{\partial x} - \frac{d}{dt} \left(\frac{\partial L}{\partial x'} \right) \right] \eta(t) dt = 0.$$

Y finalmente, aplicando el Teorema Fundamental del Cálculo Variacional (TFCV), tenemos que se satisfacen las ecuaciones de *Euler-Lagrange (EL)*

$$\frac{\partial L}{\partial x} - \frac{d}{dt} \left(\frac{\partial L}{\partial x'} \right) = 0$$

Para mas de una coordenada espacial, estas ecuaciones se suelen escribir como

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}_i} \right) = \frac{\partial L}{\partial x_i}. \quad (2.24)$$

Ahora bien, si definimos las cantidades dinámicas, el *momento generalizado*

$$\mathbf{p}_i = \frac{\partial L(x_i(t), \dot{x}_i(t))}{\partial \dot{x}_i} \quad (2.25)$$

y la *fuerza generalizada*

$$F_i = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}_i} \right) - \frac{\partial L}{\partial x_i}. \quad (2.26)$$

Usando la definición de momento generalizado en la de fuerza generalizada, se tiene

$$\frac{\partial L}{\partial x_i} = \dot{x}_i - F_i,$$

expresión que a su vez podemos sustituir en la variación del Lagrangiano

$$dL = \left[\frac{\partial L}{\partial x_i} dx_i - \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}_i} d\dot{x}_i \right) \right] + \frac{\partial L}{\partial t} dt$$

para obtener

$$dL = \left[(\dot{x}_i - F_i) dx_i - \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}_i} d\dot{x}_i \right) \right] + \frac{\partial L}{\partial t} dt.$$

Reescribiendo la expresión anterior, podemos llegar a

$$(\mathbf{p}_i \dot{x}_i - L) = [(F_i - \dot{\mathbf{p}}_i) dx_i + \dot{x}_i d\mathbf{p}_i] - \frac{\partial L}{\partial t} dt.$$

Definimos ahora otra cantidad dinámica llamada Hamiltoniano $H : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ como

$$H = \mathbf{p}_i \dot{x}_i - L, \quad (2.27)$$

con lo cual

$$H = [(F_i - \dot{\mathbf{p}}_i) dx_i + \dot{x}_i d\mathbf{p}_i] - \frac{\partial L}{\partial t} dt = \left[\frac{\partial H}{\partial x_i} dx_i + \frac{\partial H}{\partial p_i} dp_i \right] + \frac{\partial H}{\partial t} dt.$$

Asociando términos en la última ecuación del párrafo anterior, obtenemos las *ecuaciones de Hamilton (EH)*

$$\begin{cases} \frac{\partial H}{\partial x_i} = F_i - \dot{p}_i \\ \frac{\partial H}{\partial p_i} = \dot{x}_i \\ \frac{\partial H}{\partial t} = -\frac{\partial L}{\partial t} \end{cases} .$$

Como ejemplo, supongamos que tenemos el Lagrangiano

$$L(x, v) = \frac{m|v^2|}{2} - U(x).$$

Entonces $\frac{\partial L}{\partial x} = \nabla U$, $\frac{\partial L}{\partial v} = mv$, con lo que las ecuaciones de EL se escriben como

$$m\ddot{x} = -\nabla U,$$

que de hecho es la segunda Ley de Newton para una fuerza conservativa. Por otro lado, usando la definición de momento generalizado tenemos $\mathbf{p} = mv$, que nuevamente coincide con el momento newtoniano.

De esta forma, el hamiltoniano correspondiente es

$$H = \frac{\mathbf{p}^2}{2m} + U$$

y por tanto las ecuaciones de Hamilton son

$$\begin{cases} \dot{x} = \frac{\mathbf{p}}{m} \\ \dot{\mathbf{p}}_i = -\nabla U \end{cases} .$$

Ahora estamos por fin en posición de enunciar el enunciado teórico más importante de la teoría de control óptimo, el *Principio Maximal de Pontryagin (PMP)*.

Para introducir este principio, pensemos que $\tilde{\alpha}$ es un control óptimo, entonces existe una función \tilde{p}_i llamada *condición de máximo* que representa al estado adjunto, que satisfecerá el PMP. Esta condición de máximo, es un multiplicador de Lagrange que surge de resolver el problema de control (2.8).

En este punto hay que recordar que los multiplicadores de Lagrange surgen del problema de encontrar \tilde{x} que maximiza una función suave $f : \mathbb{R}^n \rightarrow \mathbb{R}$, sujeta a la condición $\tilde{x} \in R$; $R \doteq \{x \in \mathbb{R}^n \mid g(x) \leq 0\}$ para $g : \mathbb{R}^n \rightarrow \mathbb{R}$. Este problema de maximización, implica que existen dos escales no nulos simultáneamente μ, λ tales que

$$\mu \nabla f(\tilde{x}) = \lambda \nabla g(\tilde{x}).$$

Es conveniente además escribir al Hamiltoniano en terminología de control

$$H(x_i, p_i, \alpha_i) \doteq f_i(x_i, \alpha_i)p_i + r(x_i, \alpha_i); \quad (x_i, p_i \in \mathbb{R}^n, \alpha_i \in \mathcal{A}).$$

El **Principio Maximal de Pontryagin**, establece que siendo $\tilde{\alpha}$ un control óptimo de (2.8)-(2.9) y siendo \tilde{x} su correspondiente trayectoria; entonces existe una función $\tilde{p}_i : [0, T] \rightarrow \mathbb{R}$ tal que se satisfacen las ecuaciones de Hamilton

$$\begin{cases} \dot{\tilde{x}}_i(t) = \frac{\partial H(\tilde{x}_i, \tilde{p}_i, \tilde{\alpha}_i)}{\partial p_i} \\ \dot{\tilde{p}}_i(t) = \frac{\partial H(\tilde{x}_i, \tilde{p}_i, \tilde{\alpha}_i)}{\partial x_i} \end{cases}, \quad (2.28)$$

así como la condición maximal

$$H(\tilde{x}_i, \tilde{p}_i, \tilde{\alpha}_i) = \max_{\alpha \in \mathcal{A}} \{H(\tilde{x}_i, \tilde{p}_i, \alpha_i)\}; \quad 0 \leq t \leq \tilde{t}, \quad (2.29)$$

una condición de mapeo constante

$$t \mapsto H(\tilde{x}_i, \tilde{p}_i, \tilde{\alpha}_i), \quad (2.30)$$

y una condición de transversalidad

$$\tilde{p}(T) = \nabla g(\tilde{x}(T)). \quad (2.31)$$

Una demostración del PMP se puede consultar en el apéndice correspondiente, por lo que en esta sección optamos por ejemplificar una de las formas de usar PMP. Para ello regresaremos al ejemplo de producción visto en (§2.4.2), donde el sistema evoluciona según

$$\begin{cases} \dot{x}(t) = k\alpha(t)x(t) \\ x(0) = x^0 \end{cases},$$

con el funcional de recompensa

$$P[\alpha(\cdot)] = \int_0^T (1 - \alpha(t))x(t)dt.$$

Siendo³

$$f(x, \alpha) = \alpha x, \quad g \equiv 0, \quad r(x, \alpha) = (1 - \alpha)x;$$

tenemos que el Hamiltoniano de control es

³Por simplicidad, en el ejemplo no escribimos la tilde, $\tilde{x}, \tilde{p}, \tilde{\alpha} \rightarrow x, p, a$

$$H(x, p, \alpha) = f(x, \alpha)p + r(x, \alpha) = px\alpha + (1 - \alpha) = x + \alpha x(p - 1),$$

las ecuaciones de Hamilton

$$\begin{cases} \dot{x}(t) = \alpha x \\ \dot{p}(t) = -1 - \alpha(p - 1), \end{cases}$$

la condición de transversalidad

$$p(t) = \frac{\partial g}{\partial x} = 0$$

y la condición maximal

$$H(x, p, \alpha) = \max_{0 \leq \alpha \leq 1} \{x + \alpha x(p - 1)\}.$$

De la condición maximal, se deduce que

$$\alpha = \begin{cases} 1; & p \geq 1 \\ 0; & p \leq 1 \end{cases},$$

conocido α podemos calcular p usando la segunda ecuación de Hamilton con la condición de transversalidad. Como $p(T) = 0$, por continuidad tendremos que en las cercanías de T $p \leq 1$, lo cual implica $\alpha = 0$ y sustituyendo en la segunda ecuación de Hamilton obtenemos que $\dot{p} = -1 \Rightarrow p = T - t$. De forma similar se puede concluir que lejos de T , $\alpha = 1$ con lo que sustituyendo nuevamente en la ecuación de Hamilton, tenemos que $\dot{p} = p \Rightarrow p = e^{T-1-t}$.

Restaurando la notación tildada, tenemos que

$$\tilde{\alpha} = \begin{cases} 1; & 0 \leq t \leq \tilde{t} \\ 0; & \tilde{t} \leq t \leq T \end{cases}$$

para el tiempo crítico $\tilde{t} = T - 1$.

Hasta aquí podríamos sentirnos satisfechos con las herramientas matemáticas mostradas, pues no solo son de belleza teórica sino que tienen un potencial de aplicación muy grande, como ejemplificaremos en la siguiente sección.

§2.5. Ejemplos de control en Ciencias de la Tierra.

Dado que el objetivo principal de este trabajo es la aplicación de la teoría de control moderno en la gestión óptima del agua subterránea, decidimos ejemplificar los conceptos revisados en las secciones anteriores con ejemplos en otras áreas de las ciencias de la Tierra.

Control de incendios forestales

México y Centro América son ricos en biodiversidad y sustentan unos 7.3×10^7 ha forestales, incluyendo ecosistemas adaptados al fuego o influidos por éste. Sin embargo, debido al exceso de fuego con bases de pobreza y falta de cultura forestal, pues el 43 % de las causas se relacionan con actividades agropecuarias, un promedio de 21,000 incendios afectan cada año 677,000 ha en la región. Durante 1998, unos 56,731 incendios afectaron 2,330,000 ha. Los incendios forestales contribuyen a la deforestación y a la contaminación, así como al cambio climático global, entre otros impactos negativos. Se cuenta con unos 17500 combatientes permanentes en la región, y con diferentes niveles de recursos y tecnologías para las distintas tareas de manejo del fuego. En general, los recursos para la lucha contra los incendios forestales en la región son insuficientes, si bien se han presentado avances importantes en muchos casos, con relevante apoyo de varios países. No obstante, el escenario futuro es de más incendios. Con lo cual surge la necesidad de estudiar esta problemática desde la perspectiva del control óptimo.

En este ejemplo consideraremos una aproximación al control de incendios forestales usando inclusiones diferenciales[20]. Una inclusión diferencial es una generalización del concepto de EDO del tipo

$$\dot{x} = \frac{dx}{dt}(t) \in F(t, x(t)).$$

El modelo presentado busca describir la forma en que un tomador de decisiones podría diseñar bloqueos efectivos en tiempo real contra un agente indeseado que se dispersa. La región alcanzada por el fuego $R(t) \subset \mathbb{R}^2$, dentro de la cual se busca una estrategia óptima que consiste en hallar la mejor trayectoria γ de construcción de la barrera contrafuego, así como la velocidad σ a la que debe construirse. Esta formulación del problema resulta ser bastante compleja y sus detalles técnicos van más allá del interés de este trabajo, sin embargo describiremos uno de sus resultados más claros.

Considerando que el fuego se esparce a partir de una fuente puntual en todas direcciones y a la misma velocidad, se puede demostrar que la solución óptima es una espiral creciente.

Para determinar que tan grande debe ser la velocidad de construcción de la barrera σ ,

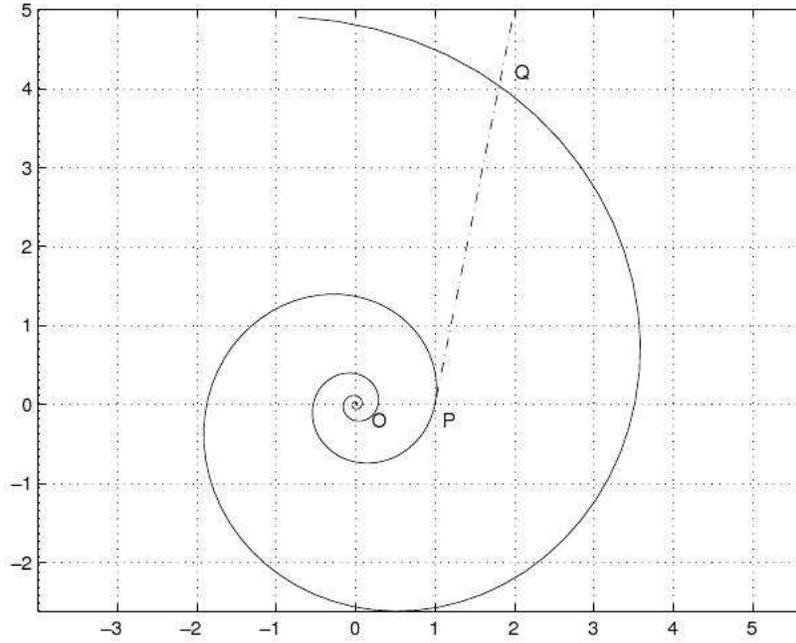


Figura 2.12: Espiral definida por ψ para $\lambda = 2$

recurrimos a la ecuación de una espiral arbitraria en polares $r = e^{\lambda\theta}$ de tal forma que podemos definir

$$S^\lambda \doteq \{(e^{\lambda\theta} \cos(\theta), e^{\lambda\theta} \sin(\theta)), \theta \in \mathbb{R}\} \subset \mathbb{R}^2.$$

Comencemos de un punto arbitrario, digamos $P = (1, 0)$, desde el cual se tiende una tangente que intersecta a S^λ en un punto Q que nos permite definir

$$\psi(\lambda) = \frac{[\text{logitud del arco de espiral entre } 0 \text{ y } P]}{[\text{logitud del arco de espiral entre } 0 \text{ y } P] + [\text{longitud del segmento } PQ]}.$$

Como moverse a lo largo de un segmento de arco entre PQ es evidentemente una trayectoria más larga que una línea recta entre ambos puntos, es claro que $\psi(\lambda) > 1$ para todo $\lambda > 0$. Más aún, como

$$Q = (e^{\lambda\Theta} \cos(\Theta), e^{\lambda\Theta} \sin(\Theta)),$$

entonces $\Theta(\lambda)$ queda definida implícitamente por la relación

$$e^{\lambda\Theta} \cos(\Theta) - 1 = \lambda e^{\lambda\Theta} \sin(\Theta)$$

con lo cual se puede demostrar que $\psi(\Theta) = \frac{1}{\cos(\Theta)}$, función que es continua y admite un

máximo en la parte positiva del eje, digamos $\tilde{\sigma} = \max_{\lambda > 0} \psi(\lambda)$. Resolviendo numéricamente, se obtiene que

$$\tilde{\sigma} = 2.614430844 \Rightarrow \tilde{\lambda} = 0.279949878.$$

Lo que nos indica que efectivamente en un caso isotrópico, si la velocidad de construcción de la barrera es mayor que $\tilde{\sigma}$ entonces, el fuego quedará bien contenido.

Capítulo 3

Control óptimo en agua subterránea

§3.1. Modelación del agua subterránea

En esta sección seguiremos de cerca el la aproximación usada en [43, 72], para posteriormente usar sus resultados generales para la deducción de la ecuación de flujo subterráneo.

En su lucha por la supervivencia, el Hombre tiene una extraordinaria ventaja al poder retener grandes cantidades de información, ya sea en su memoria individual o colectiva. Esta información de los hechos ocurridos en el pasado, nos permite construir una historia coherente que a su vez da lugar de forma natural a la predicción. Justamente esta capacidad de predicción forma el núcleo de la ciencia moderna. En este sentido, cuando hablamos de predicción científica en la actualidad, se tiene que hablar casi necesariamente de la modelación matemática y computacional [25].

Ahora bien, para el Hombre moderno existen muchos sistemas sobre los cuales quisiéramos tener una buena capacidad predicativa e incluso sobre los cuales buscamos ejercer algún tipo de control. Los sistemas de interés en la actualidad, son tan variados como los sistemas biológicos, físicos, sociales, económicos, tecnológicos o incluso combinaciones de ellos. Además de esta diversidad, conforme la sociedad ha ido desarrollándose, estos sistemas han aumentando su complejidad a tal grado que a primera vista podría parecernos imposible su modelación, menos aun su control. No obstante lo anterior, en sus cursos de Modelación de sistemas terrestres, el Dr. Herrera del Instituto de Geofísica de la UNAM, afirma que con el uso del paradigma científico moderno de *claridad, generalidad y sencillez*, hemos desarrollado una metodología lo suficientemente robusta para dar solución a un número realmente grande de problemas de modelación. [44]

La metodología matemática moderna ha alcanzado un grado tal de desarrollo que si nos enfocamos en solo una clase de sistemas: los sistemas físicos macroscópicos, podemos modelar casi cualquier problema utilizando la *Mecánica de los medios deformables* y una sola ecuación

diferencial, *la ecuación de balance*. Aunque la capacidad para formular modelos matemáticos de sistemas complicados y de gran diversidad, es sin duda una contribución fundamental para el avance de la ciencia y sus aplicaciones; tal contribución sería incompleta y de poca aplicación real sin su complemento computacional. Esto es así por la naturaleza misma de la teoría matemática en la cual se fundamenta, las Ecuaciones Diferenciales Parciales (EDP) para las cuales en la mayoría de los casos no hay soluciones analíticas y por tanto su solución solo puede obtenerse mediante el uso de métodos numéricos.

§3.1.1. Mecánica de los medios deformables.

La mecánica de los medios deformables o continuos, como también se les conoce, trata sobre los cambios de las posiciones relativas de los puntos que conforman a un cuerpo sólido o fluido, cuando éste se deforma por la acción de un agente físico sobre él. En esta definición, estamos de manera implícita definiendo también a un cuerpo deformable, sin embargo para clarificar el asunto definiremos a su contra parte los cuerpo rígidos. Los cuerpos rígidos son aquellos cuyos distancias intermoleculares o interatómicas, permanecen invariantes sin importar el agente físico que pudiera actuar sobre ellos. La definición anterior implica que dado un estímulo externo aplicado sobre el cuerpo rígido, éste debería propagarse por el mismo a una velocidad infinita lo cual va en contra de lo observado hasta ahora, donde no se ha medido velocidad alguna mayor a la de la luz. En un sentido riguroso hay muy buenas razones para suponer que de hecho la velocidad de la luz es una cantidad límite e invariante para este universo. Por otro lado, incluso antes de Maxwell, se habían observado muchas semejanzas entre el comportamiento de la luz y los fluidos, sin embargo un tratamiento puramente mecanista de la luz no puede explicar todos los fenómenos relacionados con la misma. Finalmente, las partículas elementales o sub-elementales tampoco son susceptibles de ser estudiadas bajo esta teoría pues caen en el dominio de la mecánica cuántica. Lo cual nos deja concluir que dejando de lado la idealización (inexistente) del cuerpo rígido, la radiación y los sistemas cuánticos, todo lo demás puede muy bien ser considerado un cuerpo deformable.

Dicho esto, hay que aclarar entonces aún cuando un cuerpo deformable es ante todo un sistema macroscópico, si se hacen sucesivas subdivisiones del mismo, cada parte tendrá las mismas propiedades físicas, es decir que se trata de un sistema continuo. Por supuesto dichas subdivisiones nunca deberán ser tales que se alcance la escala atómica y con ello surjan fenómenos cuánticos. Es decir, cada partición del cuerpo deformable original deberá contener un número suficiente de moléculas para que su estadística pueda ser descrita por una distribución del tipo Maxwell-Boltzman. Bajo esta óptica, las propiedades intensivas de los cuerpos deformables son invariantes de escala, lo cual permite inferir el comportamiento de un sistema muy grande a partir del estudio de elementos muy pequeños del mismo, vía

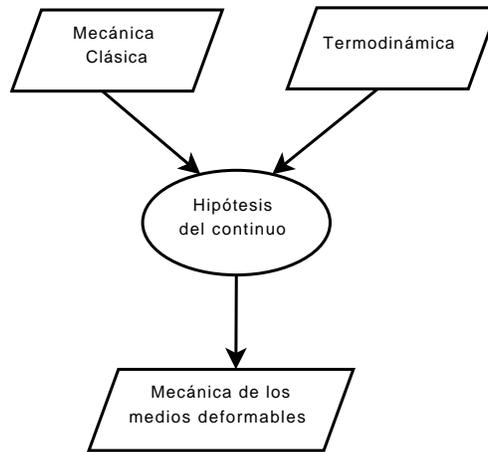


Figura 3.1: Esquema de la construcción lógica de la Mecánica de los medios deformables

un proceso de integración. A esto se le conoce como hipótesis del continuo.

El esquema fundamental de esta teoría fue propuesto por primera vez por Sir. George Stokes, quien utilizó como pilares para su construcción teórica, la mecánica de Newton y la Termodinámica de Gibbs.

Con la hipótesis del medio continuo, las leyes fundamentales de la mecánica y termodinámicas clásicas adquieren la forma de un sistema de ecuaciones diferenciales llamadas ecuaciones de balance, que describen el estado dinámico de elementos diferenciales de un cuerpo deformable. Se trata de un sistema de ecuaciones diferenciales acopladas: la de balance de masa, tres de balance de momento y una de balance de energía. De esta forma se puede en principio conocer el estado dinámico de cualquier cuerpo deformable, homogéneo, con una sola especie química, en el que haya un algún flujo y algún proceso termodinámico. Si el cuerpo deformable estuviera conformado por mas especies, flujos o procesos, se tendrían que incorporar nuevas ecuaciones de balance.

Ahora bien, para que estas ecuaciones de balance puedan volverse operativas, deben ser alimentadas con información específica del sistema de estudio. Es necesario pues especificar las características constitutivas del sistema, los agentes físicos internos que mantienen su estructura, su estado termodinámico y las fuerzas externas que actúan sobre él. Esta información se incorpora al cuerpo de las ecuaciones de balance mediante expresiones matemáticas que reciben el nombre de ecuaciones constitutivas. Las ecuaciones constitutivas son fórmulas empíricas, es decir que provienen de correlaciones experimentales y no forman parte de la estructura epistémica de la mecánica de los medios deformables. No obstante, sin las ecuaciones constitutivas, la mecánica de los medios deformables no podría resolver

ningún problema real. Un ejemplo de ley constitutiva, que de hecho no se ha podido deducir de primeros principios, es la ley de Fourier

$$q_i = k_{x_i x_j} \frac{\partial T}{\partial x_j}$$

donde q_i es el vector de flujo de calor, $k_{x_i x_j}$ es el tensor de conductividad térmica del cuerpo y $\frac{\partial T}{\partial x_j}$ es el gradiente que produce el flujo q_i .

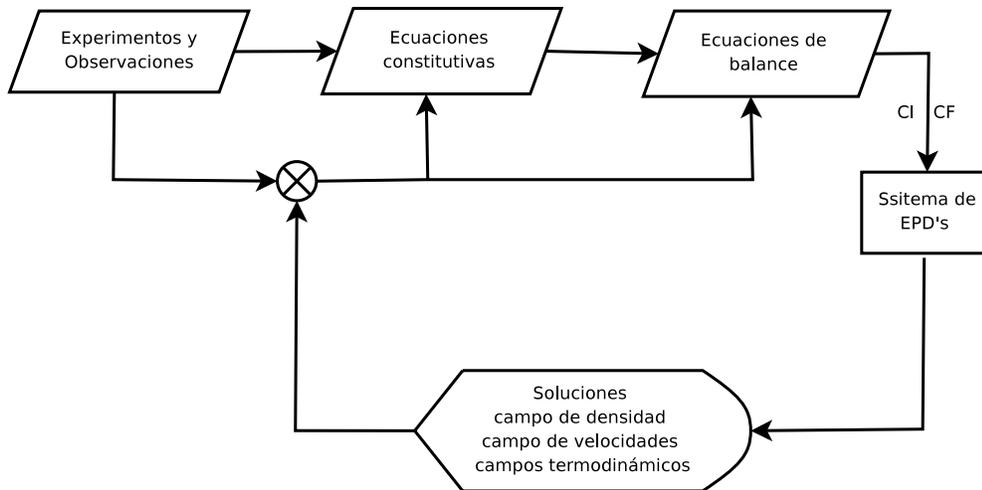


Figura 3.2: Diagrama de flujo para volver operativas a las ecuaciones de balance

Una vez que se han integrado las ecuaciones de balance junto con las constitutivas, es necesario imponer las condiciones iniciales o de frontera que terminan de definir el problema. Mediante esta estrategia mostrada en la Fig 3.2, arribamos por fin a la solución del problema, la cual esta conformada (para el caso homogéneo, uni-especie) de cinco campos, dos de ellos el de densidad de masa y Temperatura (aunque también puede ser el potencial químico), nos definen el estado termodinámico del cuerpo; mientras que el tercero, el campo de velocidades, define el estado dinámico.

Cinemática de los medios continuos

En la teoría de los sistemas deformables, los cuerpos llenan de forma continua el espacio físico que ocupan, de tal forma que en cada punto hay una y solamente una partícula. Así, podemos definir a un cuerpo como un conjunto de partículas que en cualquier instante dado ocupa un subdominio del espacio Euclidiano tridimensional. Denotaremos por $B(t)$ a la región del espacio ocupada por el cuerpo \mathbf{B} , en el tiempo t , con t un real cualquiera.

De acuerdo con la hipótesis del continuo, en cualquier tiempo $t \in (-\infty, \infty)$ y en cada punto $x \in B$ de la región ocupada por el cuerpo, hay una y sólo una partícula del cuerpo; por

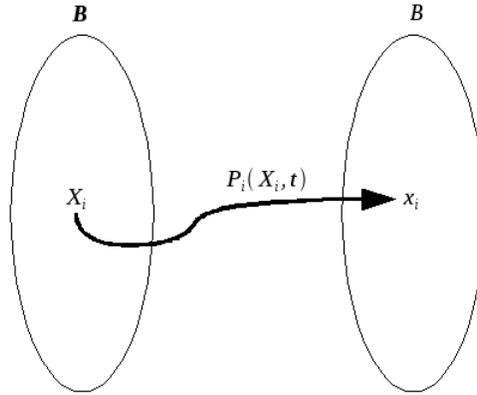


Figura 3.3: Movimiento de las partículas del cuerpo **B**

lo tanto, un primer problema de la cinemática de los sistemas continuos consiste en establecer un procedimiento para identificar a las partículas cuando están en movimiento en el espacio físico.

Sea $X_i \in B$, una partícula y $P(X_i, t)$ el vector de la posición que ocupa como en la Fig. §3.1.1. Una forma, pero no la única, de identificar dicha partícula es asociándole la posición que ocupa en el espacio físico en un instante determinado. Definamos entonces al vector de coordenadas materiales $X = (x_1, x_2, x_3)$ de tal forma que $X = P(X_i, 0)$. Como desde luego, el tiempo inicial puede ser cualquier otro, si así se desea, la definición anterior es general.

De esta forma el subdominio del espacio físico ocupado por el cuerpo $B(t)$ se puede definir como

$$B(t) = \{x_i \in \mathbb{R}^3 | X_i \in B \text{ tal que } x_i = P(X_i, t)\},$$

con lo que tenemos dos formas de describir el movimiento del cuerpo **B**, fijando t y dejando variar X_i o fijando X_i y dejando variar t . Si en particular se fija X_i en $P(X_i, t)$ y se varía el tiempo, se obtiene la trayectoria de la partícula. De esta forma se construye de forma natural la velocidad física de la partícula como

$$V_i = \frac{\partial P(X_i, t)}{\partial t}, \quad (3.1)$$

con lo cual se pueden extender todos los conceptos de la cinemática del cuerpo rígido a la cinemática de los medios deformables.

Propiedades Intensivas y Extensivas

Ahora bien, para completar nuestra dinámica de los medios deformables, es necesario poder definir las distintas propiedades intensivas y extensivas del cuerpo. Una propiedad intensiva es una propiedad que no depende de la extensión del cuerpo, como lo es la velocidad o el punto de ebullición.

Como mencionamos antes, hay dos formas de representar la dinámica del cuerpo, una en donde se fija la posición y otra en donde se fija el tiempo. Estas dos representaciones que reciben los nombres de Euleriana y Lagrangiana; son formalmente equivalentes, pero suelen tener campos de aplicación diferentes. Frecuentemente la representación Lagrangiana se usa en el estudio de los sólidos, mientras que la Euleriana se aplica más extensamente en el estudio de los fluidos.

Consideremos una propiedad intensiva I escalar representada por la función $\phi : B \rightarrow \mathbb{R}$ para todo $t \in \mathbb{R}$, la cual indica el valor que toma la propiedad intensiva en el tiempo t , en la partícula X_i . Entonces, $\phi(X_i, t)$ se llama representación lagrangiana de la propiedad intensiva I . Si ahora, para cada $t \in \mathbb{R}$ la misma propiedad intensiva I se representa por la función $\psi : B(t) \rightarrow \mathbb{R}$, la cual indica el valor que toma la propiedad en la posición x_i al tiempo t , dicha función $\psi(x_i, t)$ es la representación euleriana de I . Nótese que ambas representaciones están relacionadas por la identidad

$$\phi(X_i, t) = \psi(P(X_i, t), t).$$

De esta forma, la velocidad definida en (3.1) es la representación lagrangiana de la propiedad intensiva velocidad, es decir, la velocidad lagrangiana. De lo anterior queda claro que para una propiedad intensiva I cualquiera, en su representación lagrangiana, la derivada respecto del tiempo es simplemente $\partial_t \phi$. Por su parte, en la representación euleriana la misma derivada respecto del tiempo, conocida como derivada material, sería

$$\frac{D\psi}{Dt} = \frac{\partial \psi}{\partial t} + v_i \frac{\partial \psi}{\partial x_i} \quad (3.2)$$

con $v_i = V_i(P^{-1}(x_i, t), t)$ como la velocidad euleriana.

De forma análoga la aceleración tendrá una representación lagrangiana

$$\frac{\partial V_i}{\partial t} = \frac{\partial^2 P(X_i, t)}{\partial t^2},$$

y una euleriana

$$\frac{Dv_i}{Dt} = \frac{\partial v_i}{\partial t} + v_i \frac{\partial v_i}{\partial x_i}.$$

Finalmente, no sólo nos interesa conocer las propiedades intensivas del cuerpo, sino también las extensivas como la masa, las cuales pueden definirse a través de las intensivas como

$$E(B, t) = \int_{B(t)} \psi(x_i, t) dx_i. \quad (3.3)$$

Hay que enfatizar en este punto, que la razón por la que introdujimos los conceptos anteriores es por que el modelo general de los sistemas continuos se formula en términos de ecuaciones de balance de propiedades extensivas, como se verá a continuación.

Como se mostró en la Fig.3.2 en la estructura de solución de la mecánica de los medios deformables, es necesario construir las ecuaciones de balance. Estas ecuaciones de balance operan sobre propiedades extensivas tales como la masa o el momento y se formulan de forma general por la ecuación de balance global:

$$\frac{dE}{dt} = \int_B g(x_i, t) dx_i + \int_{\partial B} q(x_i, t) dx_i + \int_{\Sigma B} g_\Sigma(x_i, t) dx_i. \quad (3.4)$$

Donde la primera integral del lado derecho de la ecuación es el término de fuentes-sumideros, la segunda integral es cuanto atraviesa las fronteras del cuerpo, mientras que la tercera integral es la producción en la superficie del cuerpo. Ahora bien, definiendo a $q(x_i, t) = \Phi_i(x_i, t) n_i(x_i, t)$, donde Φ_i es un campo vectorial y n_i es la normal exterior a ∂B , se puede demostrar que la ecuación de balance global es equivalente a una ecuación de balance local en términos de la variable intensiva asociada, más una ecuación de salto dadas respectivamente por las ecuaciones

$$\partial_t \psi + \partial_{x_i} (v_i \psi) = \partial_{x_i} \Phi_i + g, \quad (3.5)$$

$$(\psi(v_i - v_i^\Sigma) - \Phi_i)_j n_j = g_\Sigma. \quad (3.6)$$

§3.2. El flujo subterráneo.

Como se dijo en la subsección anterior, para poder utilizar la mecánica del medio continuo, necesitamos conocer las ecuaciones constitutivas. Cuando hablamos de agua subterránea, nos referimos al agua que habiéndose infiltrado por el suelo ha llegado a saturar por completo el medio y comienza a fluir. Hablamos entonces de un flujo de agua a través de un medio poroso.

Entre 1834 y 1840 la municipalidad de Dijon, Francia, encomendó a un joven ingeniero

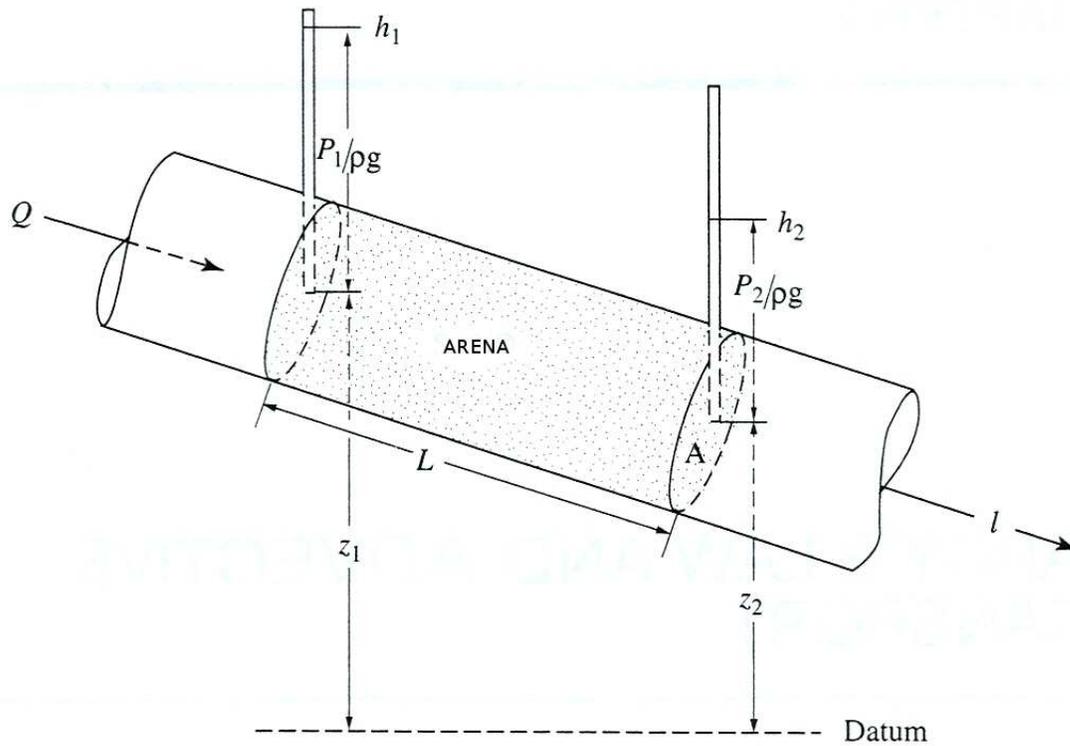


Figura 3.4: Dispositivo experimental de Darcy.

francés el diseño y construcción del sistema de abastecimiento de agua potable de la ciudad. Para 1847, el agua entubada llegaba ya a todos los pisos de todos los edificios de Dijon, transformando así a esta ciudad en la segunda ciudad europea en lo que se refiere a abastecimiento de agua, después de Roma. Durante estos años de trabajo en Dijon, Darcy se topo con la necesidad de entender como fluye el agua a través de un medio poroso, para lo cual construyo un dispositivo experimental como el de la Fig3.4

En 1856, publica un tratado sobre las fuentes públicas de Dijon, en el cual aparece la fórmula que desde entonces lleva su nombre. Haciendo referencia a la Fig 3.4, la ley de Darcy se puede escribir como

$$Q = -kA \frac{dh}{dl},$$

donde h llamada carga hidráulica, mide la energía potencial del sistema; k es la conductividad hidráulica y A el área de la sección transversal. Sin embargo, suele escribirse en términos de la velocidad del fluido visto desde el exterior, $q = \frac{Q}{A}$. Si además consideramos un

flujo en tres dimensiones y permitimos que la conductividad hidráulica también tenga tres componentes, entonces tenemos que

$$q_i = -k_{x_i x_j} \frac{\partial h}{\partial x_j}. \quad (3.7)$$

Gracias a la hipótesis del continuo y siguiendo el desarrollo de la sección anterior, podemos asociarle a cada punto del espacio una propiedad intensiva $B_i = B_i(t, x_i(t))$, para la cual se puede escribir una ecuación local como la 3.5. Escogemos ésta propiedad intensiva como la presión hidrostática y suponiendo que la formula $P = \rho gh$ es válida; entonces como $h = \frac{P}{\rho g}$ y debido a que la presión es energía por unidad de volumen, podemos concluir que la energía del sistema está caracterizada implícitamente por la carga hidráulica. De esta forma la energía almacenada en un elemento de longitud $[a, b]$ esta dada por

$$\int_a^b h(x, t) dx.$$

De la misma forma, el cambio de dicha energía en el tiempo es

$$\frac{d}{dt} \left[\int_a^b h(x, t) dx \right].$$

Lo cual nos permite escribir la ecuación de balance global del tipo 3.4

$$\frac{d}{dt} \left[\int_a^b h(x, t) dx \right] = \text{flujo de energía entrante} - \text{flujo de energía saliente}.$$

Pero como toda la energía del sistema esta dada por flujos de masa, entonces los flujos de energía son equivalentes a los de masa, con lo que podemos relacionar ésta ecuación de balance de energías con una de balance de masas, para escribir

$$\frac{d}{dt} \left[\int_a^b h(x, t) dx \right] = \frac{q_e}{A} - \frac{q_s}{A}.$$

Ordenando términos y usando el Teorema Fundamental del Cálculo la ecuación anterior se escribe como

$$\frac{d}{dt} \left[\int_a^b h(x, t) dx \right] = -\frac{1}{A} \int_a^b \partial_x q(x, t),$$

o bien

$$\int_a^b \partial_x h(x, t) dx = -\frac{1}{A} \int_a^b \partial_x q(x, t),$$

y por tanto

$$A \partial_x h = \partial_x q.$$

Introduciendo ahora la ley constitutiva correspondiente, la ley de Darcy, tenemos que para una dimensión, el flujo de agua subterránea estará modelada por

$$A \partial_t h = \partial_{x_i} [k_{x_i x_j} \partial_{x_j} h].$$

Generalizando el procedimiento mostrado, suponiendo una área transversal unitaria y que existen términos fuente o sumideros W , así como un término almacenamiento específico del medio poroso S_s ; se llega finalmente a la Ecuación de Flujo Subterráneo

$$S_s \partial_t h = \partial_x [k_{xx} \partial_x h] + \partial_y [k_{yy} \partial_y h] + \partial_z [k_{zz} \partial_z h] + W. \quad (3.8)$$

En este punto, vale la pena remarcar que como todas las leyes constitutivas, la Ley de Darcy tiene un origen experimental y su aplicación no es universal. Por ejemplo si la velocidad del fluido es suficientemente grande, el medio es muy anisotrópico (medio muy fracturado), el acuífero es del tipo no confinado, o bien cuando el fluido no es newtoniano; entonces a la ley de Darcy se le tienen que incorporar términos no lineales. Todavía más, estas modificaciones pueden dar lugar a que se modifique por completo la forma de la EDP dando lugar a lo que se conoce como ecuación del medio poroso. Sin embargo en, muchos sistema de agua subterránea, la aproximación a primer orden de Darcy así como la Eq(3.8) reproducen con suficiente exactitud los datos de campo.

§3.2.1. Solución numérica

En esta etapa, el modelo matemático está completo, y es aquí donde aparece una nueva dificultad: resolverlo. Debido a que solamente la forma más simple de EDP y con fronteras geoméricamente triviales son susceptibles de ser resueltas en forma analítica, en la mayoría de los casos será necesario resolver el problema de forma numérica usando computadoras. Para esto, es necesario presentar el problema de una manera puramente algebraica, comenzando con un proceso de discretización que represente a las funciones incógnitas en el continuo, como un número finito de parámetros discretos incógnita.

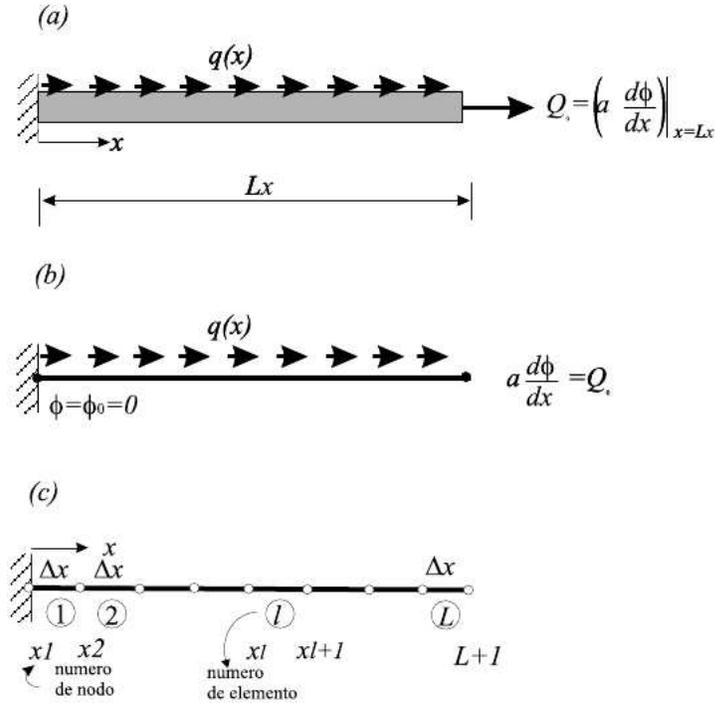


Figura 3.5: Discretización de un dominio unidimensional: (a) problema físico; (b) idealización matemática; (c) discretización por diferencias finitas

Entre los diferentes métodos de discretización (diferencias finitas, elementos finitos, o volúmenes finitos, etc), una de los más simples es el Método de Diferencias Finitas (MDF)

Comencemos por construir un malla de $L + 1$ puntos discretos igualmente espaciados x_l ($l = 0, 1, 2, \dots, L$) sobre el rango $0 \leq x \leq L_x$, con $x_0 = 0$, $x_L = L_x$, y $x_{l+1} - x_l = \Delta x$ (ver Fig.3.5). El próximo paso consiste en reemplazar aquellos términos en la ecuación diferencial que involucren diferenciación por términos que contengan operaciones algebraicas. Este proceso trae implícito una aproximación y puede efectuarse mediante la utilización de aproximaciones en diferencias finitas para las derivadas de una función. La aproximación por diferencias finitas se fundamenta en el teorema de aproximación de Taylor

$$\phi(x_{l+1}) = \phi(x_l + \Delta x) = \phi(x_l) + \Delta x \left. \frac{d\phi}{dx} \right|_{x=x_l} + \frac{\Delta x^2}{2} \left. \frac{d^2\phi}{dx^2} \right|_{x=x_l + \theta_1 \Delta x}$$

donde θ_1 es un número tal que $0 \leq \theta_1 \leq 1$. Por lo tanto $\frac{d\phi}{dx} \approx \frac{\phi_{l+1} - \phi_l}{\Delta x}$ con un error $E = \frac{\Delta x^2}{2} \left. \frac{d^2\phi}{dx^2} \right|_{x=x_l + \theta_1 \Delta x}$ que se conoce como aproximación hacia adelante. De manera similar se puede obtener la aproximación para la segunda derivada $\frac{d^2\phi}{dx^2} \approx \frac{\phi_{l+1} - 2\phi_l + \phi_{l-1}}{\Delta x^2}$.

De esta forma, si en general tenemos una EDP de segundo orden

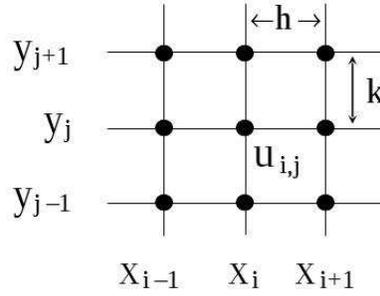


Figura 3.6: Malla rectangular uniforme

$$Au_{xx} + Bu_{xy} + Cu_{yy} + Du_x + Eu_y + Fu = G,$$

usando una generalización de la aproximación por Taylor y haciendo referencia a la Fig.3.6 se obtienen la siguientes expresiones de las derivadas parciales en diferencias finitas:

$$u_x(x_i, y_i) = \frac{u(x_i + h, y_i) - u(x_i, y_i)}{h} = \frac{u_{i+1,j} - u_{i,j}}{h} = \frac{u_{i,j} - u_{i-1,j}}{h} = \frac{u_{i+1,j} - u_{i-1,j}}{2h}$$

$$u_{xx}(x_i, y_i) = \frac{u_x(x_{i+h}, y_i) - u_x(x_i, y_i)}{h} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2}.$$

Solo para ejemplificar, una ecuación parabólica del tipo $u_y = cu_{xx}$ se escribe en diferencias finitas como

$$\frac{u_{i,j+1} - u_{i,j}}{k} = c \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2},$$

con lo cual el problema se puede escribir ya en términos puramente algebraicos.

§3.3. Manejo óptimo.

El manejo sustentable del agua subterránea se ha convertido en un tema de importancia crucial en el aseguramiento del agua en regiones donde la disponibilidad de agua superficial esta limitada. Desde hace un par de décadas, la planeación a largo plazo del uso de agua subterránea ha venido dependiendo cada vez más de la modelación computacional; la cual debe poder predecir con suficiente exactitud el comportamiento del flujo subterráneo, al tiempo que permita explorar diferentes escenarios de manejo basados en criterios como la sustentabilidad. [76, 12, 57].

Por supuesto, cualquier escenario de manejo debe ser consistente con las leyes gobernantes del sistema, por lo cual es necesario incorporar a cualquier metodología de manejo un modelo

del comportamiento físico del sistema. Bajo esta perspectiva, el término manejo implica desde una perspectiva matemática un problema de optimización con limitaciones, las cuales pueden ser impuestas por las características física del sistema, por requerimientos de los usuarios, o bien por políticas de las autoridades que manejan el recurso.

La mancuerna entre modelación y optimización ha mostrado ser muy exitosa para determinar estrategias de manejo de agua subterránea. Los modelos que simulan el flujo subterráneo son capaces de contestar preguntas tales como ¿qué pasaría si...?, donde después del si puede agregarse, poniendo sólo un ejemplo: el bombeo promedio de los pozos aumentará un X % en el siguiente año. Por su parte, las técnicas de optimización nos permitirían, siguiendo el ejemplo anterior, identificar el valor numérico del aumento promedio de los pozos: X, tal que ciertas condiciones se satisficieran. Ahora bien, una vez que el problema de optimización esta bien planteado, se tiene que aplicar una técnica matemática y computacional adecuada para resolverlo. Dichas técnicas se han diversificando a lo largo del tiempo, ofreciendo en la actualidad una amplia diversidad.

§3.3.1. Programación Lineal (PL)

La programación Lineal es una técnica de optimización que busca dar respuesta a la pregunta ¿Cuales serán los mejores controles A , tales que maximicen/minimicen la función objetivo $\mathbf{c}^T \mathbf{x}$, sujeta a las restricciones $A\mathbf{x} \leq \mathbf{b}$? Esto implica hacer un planteamiento hidrológico a través de control clásico §2.3 usando matrices de transferencia o como se conocen en hidrología, matrices de respuesta.

El alcance de esta técnica esta limitado a que tanto la función objetivo como las restricciones físicas y de manejo sean lineales. No obstante esta limitante, las técnicas de PL pueden resolver problemas de gran escala y garantizan la obtención de un óptimo global. Adicionalmente se cuenta ya con un amplio repertorio de algoritmos numéricos para llevar acabo esta labor, de esta aproximación una de las más usadas en problemas de ingeniería.

Las primeras aplicaciones fueron hechas por Lee & Aronofsky [51] quienes desarrollaron un modelo de manejo que maximizaba las ganancias de explotación. Para ello construyeron las matrices de respuesta mediante la solución analítica de la ecuación de flujo Eq (3.8).

Para 1966, Hallaji y Yazicigi [42] estaban aplicando PL para el manejo de acuíferos costeros, en un sistema de seis modelos lineales acoplados tanto en estado transitorio como estacionario; una función objetivo cuadrática para estado estacionario y cuatro tipo de restricciones: a) demanda de agua, b) limitaciones en el abatimiento, c) máximas velocidades de bombeo, d) mínimas velocidades de bombeo.

§3.3.2. Programación No Lineal (PNL)

A pesar del éxito inicial de la PL, el mundo real es casi siempre no lineal. De forma general, podemos enunciar el problema de optimización no lineal como encontrar $\max_{x \in X} f(x)$, o bien $\min_{x \in X} f(x)$, con $X \in \mathbb{R}^n$; $f : \mathbb{R}^n \rightarrow \mathbb{R}$, sujeto a $g_i(x) = 0$; $i = 1, \dots, m_1$ y $h_j(x) \geq 0$; $j = m_1 + 1, \dots, m$.

Debido a que la PNL es un área de estudio intrínsecamente difícil, su estudio se ha hecho en base a identificar tipos particulares de problemas que se pueden resolver con técnicas específicas.

Si la función objetivo f es lineal y el espacio de las restricciones g_i, h_j es politopo, el problema es de Programación lineal y puede resolverse utilizando alguno de los bien conocidos algoritmos de programación lineal.

Si g_i, h_j son lineales y la función objetivo f es cuadrática, entonces el problema se llama de Programación Cuadrática (PC). La importancia de la programación cuadrática radica no sólo en la variedad de problemas que encuentran en este modelo su marco apropiado, sino porque muchos de los algoritmos para la solución del programa general no lineal se basan en la solución de subproblemas cuadráticos. Los primeros trabajos en PNL en agua subterránea, fueron de hecho en PC en los trabajos de Maddock [54] donde minimizaba el abatimiento debido al bombeo en cada pozo. A principios de este siglo, Das y Datta [28, 29, 30] publicaron una serie de tres artículos donde atacaron con PLN el bastante difícil problema de la intrusión salina en acuíferos costeros.

El método específico para resolver un problema de PC depende de la concavidad de f . En general:

1. $f(x)$ es cóncava si $\mathbf{x}^T \mathbf{H} \mathbf{x} \leq 0$ para toda \mathbf{x} .
2. $f(x)$ es estrictamente cóncava si $\mathbf{x}^T \mathbf{H} \mathbf{x} < 0$ para toda $\mathbf{x} \neq \mathbf{0}$.
3. $f(x)$ es convexa si $\mathbf{x}^T \mathbf{H} \mathbf{x} \geq 0$ para toda \mathbf{x} .
4. $f(x)$ es estrictamente convexa si $\mathbf{x}^T \mathbf{H} \mathbf{x} > 0$ para toda $\mathbf{x} \neq \mathbf{0}$.

Si la función objetivo es cóncava (problema de maximización), o convexa (problema de minimización) y el conjunto de restricciones es convexo, entonces se puede utilizar el método general de Optimización convexa.

Existe una variedad de métodos para resolver problemas no convexos. Uno de ellos consiste en utilizar formulaciones especiales de problemas de programación lineal. Otro método implica el uso de técnicas de ramificación y poda. Esta solución es óptima, aunque posiblemente no sea única. El algoritmo puede ser parado antes, con la garantía de que la mejor solución será mejor que la solución encontrada en un porcentaje acotado.

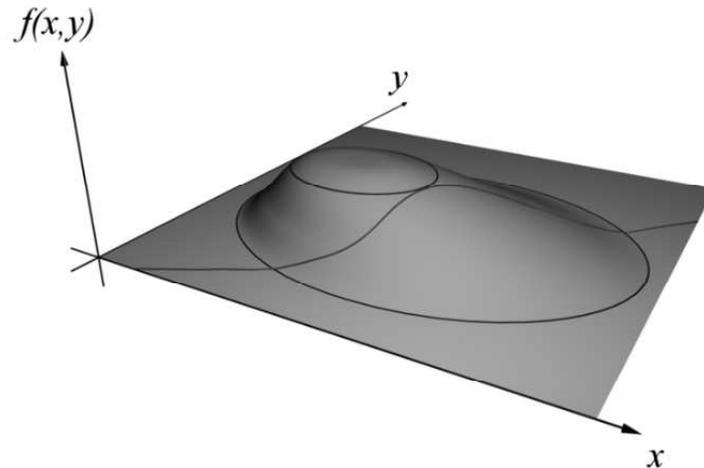


Figura 3.7: Geométricamente, lo que buscamos es el valor más grande de f pero sobre la curva g .

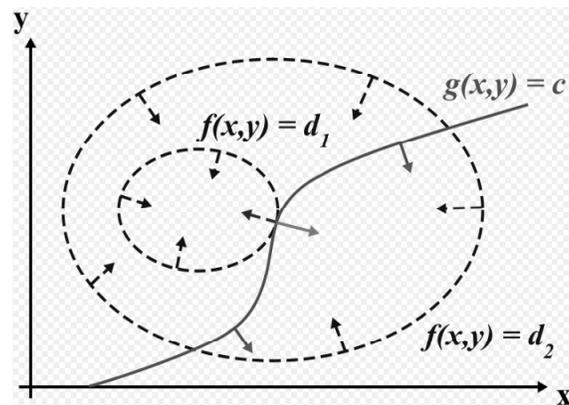


Figura 3.8: El punto buscado es aquel donde la curva g es tangente a la función objetivo f .

Una técnica tradicional de resolución para un buen número de problemas de optimización es la de los Multiplicadores de Lagrange (ML). Supongamos que queremos maximizar $f(X)$ sujeta a un conjunto de restricciones $g_k(X) = 0$, con $X \in \mathbb{R}^n$. Para un caso sencillo en \mathbb{R}^3 que corresponde a la Fig.3.7 lo que se busca es encontrar el máximo de f pero cuando nos “movemos” solo sobre la curva $g = c$. Analizado en el plano, como en la Fig.3.8, se trata de hallar el punto donde la curva g es tangencial a la función objetivo.

Para resolver este problema, introduzcamos una función llamada Función Asociada de Lagrange al problema de optimización:

$$\Lambda(X, \lambda) = f(X) + \sum_k \lambda_k g_k(X),$$

donde hemos introducido una nueva variable λ_k conocida como multiplicador de Lagrange.

Se puede demostrar entonces que los puntos que resuelve el problema de optimización son aquellos para los cuales $\nabla_{\lambda} \Lambda = 0$ lo cual ocurre si y solo si $\nabla_X f = -\sum_k \lambda_k g_k(X)$.

Una generalización de este método conocida como condiciones de Karush-Kuhn-Tucker (KKT) proporcionan las condiciones necesarias para un caso más general que admita restricciones en forma de desigualdades.

Supongamos que queremos encontrar la distribución probabilística discreta con máxima entropía,

$$f(p_1, p_2, \dots, p_n) = -\sum_{k=1}^n p_k \log_2 p_k.$$

Evidentemente, la suma de estas probabilidades debe ser exactamente igual a 1, por lo tanto nuestra restricción es $g(\mathbf{p})$, con $g(p_1, p_2, \dots, p_n) = \sum p_k$.

Podemos usar los multiplicadores de Lagrange para encontrar el punto de máxima entropía (dependiendo de las probabilidades). Para todo k desde 1 hasta n. Para ello consideremos

$$\frac{\partial}{\partial p_k}(f + \lambda(g - 1)) = 0,$$

lo que nos da

$$\frac{\partial}{\partial p_k} \left(-\sum_{k=1}^n p_k \log_2 p_k + \lambda \sum_{k=1}^n p_k - \lambda \right) = 0.$$

Derivando estas n ecuaciones, obtenemos

$$-\left(\frac{1}{\ln 2} + \log_2 p_k \right) + \lambda = 0.$$

Esto muestra que todo p_i es igual (debido a que depende solamente de λ). Usando la restricción $\sum p_k = 1$, encontramos $p_k = \frac{1}{n}$. Que es la distribución con la mayor entropía.

§3.3.3. Programación Dinámica (PD)

La programación dinámica es un conjunto de técnicas matemáticas y computacionales que tratan de responder problemas complejos rompiéndolos en varios subproblemas más sencillos de forma recursiva.

Supongamos por ejemplo que queremos llegar del nodo A al nodo J como se muestra en la Fig.3.9. Comencemos por pensar que el problema se puede completar en un número finito de pasos entre diferentes estados del sistema: en el primer estado solo el nodo A es accesible;

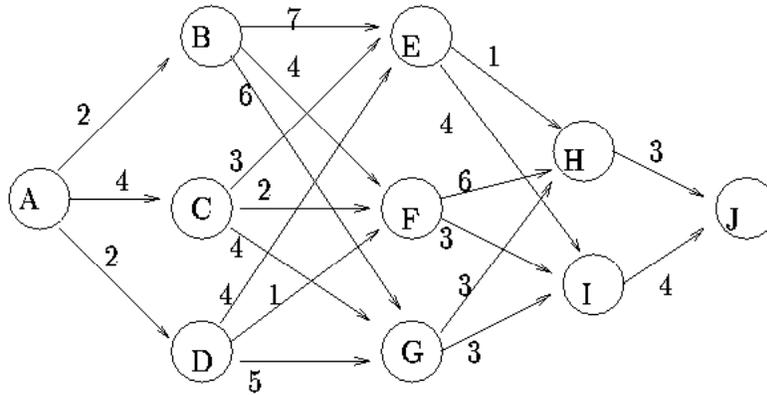


Figura 3.9: Red donde los números sobre los vértices representan distancias. Se quiere llegar de A a J por el camino de menor distancia.

en el segundo son accesibles los nodos B,C y D; en el tercero son accesibles los nodos E,F y G; en el cuarto son accesibles los nodos H e I; mientras que en el último estado solo es accesible J.

De esta forma, el problema se puede romper en varios subproblemas: pasar de un estado a otro por el camino de la menor trayectoria posible. Si S denota al nodo ocupado en el estado j y C_{SZ} es el camino entre el nodo S y el nodo Z ; entonces sea

$$f_j(S) = \min_{\text{todo } Z \text{ en } j+1} (C_{SZ} + f_j(z)),$$

la trayectoria de menor camino entre el nodo S y el J . Haciendo los cálculos, obtenemos:

Estado 5:

S_4	$C_{S_4Z_4} + f_5(S_5)$		$f_4(S_4)$	Decisión
	J			
J	0		0	—

Estado 4:

S_4	$C_{S_4Z_4} + f_5(S_5)$		$f_4(S_4)$	Decisión
	J			
H	3		3	H
I	4		4	

Estado 3:

S_3	$C_{S_3Z_3} + f_4(S_3)$		$f_3(S_3)$	Decisión
	H	I		
E	4	8	4	H
F	9	7	7	I
G	6	7	6	H

Estado 2:

S_2	$C_{S_2Z_2} + f_3(S_2)$			$f_2(S_2)$	Decisión
	E	F	G		
B	11	11	12	11	E o F
C	7	9	10	7	E
D	8	8	11	8	E o F

Estado 1:

S_1	$C_{S_1Z_1} + f_2(S_1)$			$f_1(S_1)$	Decisión
	B	C	D		
A	13	11	11	11	C o D

Las propiedades de un problema que pueda ser resuelto por PD son:

- El problema se puede dividir en estados, en los cuales se puede formular una decisión para avanzar al siguiente estado.
- Cada decisión tiene sólo un número finito de opciones.
- La decisión en cada estado es independiente de los resultados anteriores.
- Se puede formular una relación de recursividad para encontrar la decisión óptima en cada estado.
- El estado final debe poder resolverse por si mismo.

Si el costo asociado a pasar de un estado a otro; o bien el estado resultante de una decisión, se conocen solo hasta cierto grado de certeza, es decir que se expresan en términos de probabilidad, entonces estamos hablando de Programación Dinámica Estocástica.

Con una formulación más general, la Programación Dinámica Diferencial [49] se puede formular el problema general No lineal de manejo óptimo de acuíferos. Otras aplicaciones en esta vertiente se han hecho sobre todo para resolver problemas con un número muy grande de restricciones [26].

No obstante la probada utilidad de cada una de las aproximaciones resumidas en los párrafos anteriores, un problema actual sigue siendo el de optimizar funciones objetivo con muchos óptimos locales como en la Fig.3.10. Hay varios métodos que pueden resolver este tipo de problemas, pero en la actualidad tenemos además que resolver problemas multiobjetivos más complejos: muchas funciones objetivo (no lineales y con muchos óptimos locales) restringidas a su vez por muchas restricciones (No lineales e incluso en desigualdades). En este trabajo, nos enfocaremos con detalle a este tipo de problemas, pues es justamente el tipo de problemas de manejo óptimo que nos gustaría poder atacar en la realidad. En virtud de

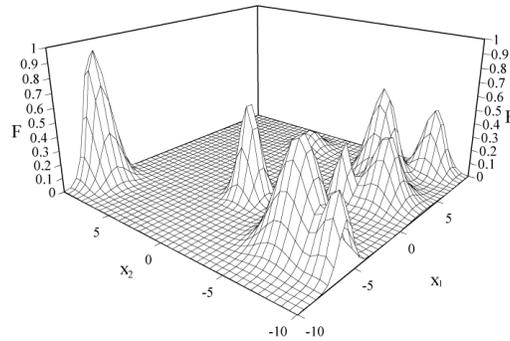


Figura 3.10: Típica función con muchos óptimos locales

lo anterior, introduciremos una aproximación nueva que esta mostrando ser extremadamente adecuada para nuestros propósitos, el Computo evolutivo (CE), que desarrollaremos en la siguiente sección.

§3.4. Computo Evolutivo.

Aunque se acepta que la programación evolutiva nació en la década de 1960 con el trabajo de Lawrence J. Fogel, sobre la evolución de máquinas de estado finitas; ya en los años 1930s, algunos investigadores comenzaron a ver a la evolución de las especies biológicas como un proceso de aprendizaje, mediante el cual la naturaleza selecciona aquellas características que resultan en que los organismos sean más aptas para sobrevivir y dejar descendencia fértil. En su libro *The Wisdom of the Body*, Walter D. Cannon [23], plantea que el proceso evolutivo es algo similar al aprendizaje por ensayo y error que suele manifestarse en los humanos. El célebre matemático inglés Alan Mathison Turing reconoció también una conexión “obvia” entre la evolución y el aprendizaje de máquina en su artículo *Computing Machinery and Intelligence* [71].

A principios de los 1960s, el biólogo Alexander S. Fraser [35, 36, 37] publicó una serie de trabajos sobre la evolución de sistemas biológicos en una computadora digital, dando la inspiración para lo que se convertiría más tarde en los Algoritmos Genéticos (AG). Su trabajo de más de 10 años en este tema se resume en un libro titulado *Computer Models in Genetics* [38]. Puede decirse que el trabajo de Fraser se anticipó a la propuesta del algoritmo genético simple de John Holland y la de la estrategia evolutiva de dos miembros de Hans-Paul Schwefel [46]; considerados como los padres de los AG.

Pero quizás fue Hans Joachim Bremermann [15] quien por primera vez vió a la evolución como un proceso de optimización. En sus trabajos realizó una de las primeras simulaciones de la evolución usando cadenas binarias que se procesaban por medio de reproducción (se-

xual o asexual), selección y mutación, en lo que sería otro claro predecesor del algoritmo genético. Bremermann y Rogson[16, 18] utilizaron una técnica evolutiva para problemas de optimización con restricciones lineales. La idea principal de su propuesta era usar un individuo factible el cual se modificaba a través de un operador de mutación hacia un conjunto de direcciones posibles de movimiento. Al extender esta técnica a problemas más complejos, Bremermann y Rogson utilizaron además operadores de recombinación especializados [19]. Bremermann fue uno de los primeros en utilizar el concepto de “población” en la simulación de procesos evolutivos, además de intuir la importancia de la co-evolución [15] (es decir, el uso de dos poblaciones que evolucionan en paralelo y cuyas aptitudes están relacionadas entre sí) y visualizar el potencial de las técnicas evolutivas para entrenar redes neuronales [17].

En 1999, por primera vez en la historia, se concedió una patente a un invento no realizado directamente por un ser humano: se trata de una antena de forma extraña, pero que funciona perfectamente en las condiciones a las que estaba destinada. No hay, sin embargo, nada injusto en el hecho de que el autor del algoritmo genético del que salió la forma de la antena se haya atribuido la autoría de la patente, pues él escribió el programa e ideó el criterio de selección que condujo al diseño patentado.

Aunque hoy es difícil distinguir las diferencias entre los distintos tipos de algoritmos evolutivos existentes, por razones sobre todo históricas, suele hablarse de tres paradigmas principales:

- Algoritmos Genéticos
- Programación Evolutiva
- Estrategias Evolutivas

Por ejemplo, la diferencia conceptual más destacable entre un AG y un programa evolutivo se presenta en la Fig.3.11

La diferencia conceptual entre los algoritmos genéticos y la programación evolutiva, estriba en que los AG tradicionales solo operan sobre cadenas binarias de datos, por lo que se requiere modificar el problema original para que la información del mismo sea susceptible de ser puesta en forma binaria. Por el contrario la PE deja a el problema original sin modificación, teniendo entonces que adecuar el algoritmo al problema específico en cuestión.

§3.4.1. Algoritmos Genéticos

Para muchos problema de optimización en aplicaciones del mundo real, no existen algoritmos deterministas suficientemente rápidos y robustos para encontrar su solución. Por el contrario, métodos pobabilísticos exhaustivos como los de tipo Monte Carlo, han probado

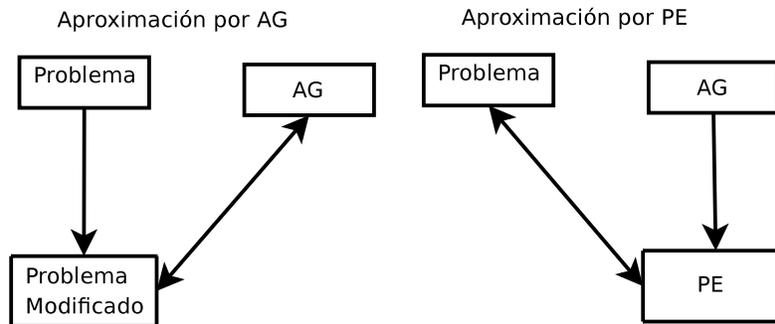


Figura 3.11: La diferencia conceptual entre Algoritmos Genéticos y Programación Evolutiva.

tener éxito en una gama amplia de problemas [31]. Sin embargo, cuando el espacio de posibles soluciones no es lo suficientemente pequeño, el tiempo y poder de cómputo que se requeriría para hallar una solución óptima es demasiado grande. Es en ese escenario donde los AG han mostrado ser sumamente adecuados [8, 10, 34]

No obstante, es importante tener en mente que los algoritmos evolutivos son técnicas heurísticas. Por tanto, no garantizan que convergerán al óptimo de un problema dado, aunque en la práctica suelen lograrlo en un tiempo promedio considerablemente menor que los algoritmos deterministas. Sin embargo, antes de decidir recurrir a los algoritmos evolutivos, debe analizarse la factibilidad de utilizar otro tipo de técnicas deterministas.

Los Algoritmos Genéticos fueron desarrollados por John H. Holland a principios de los 1960s, motivado por su interés en resolver problemas de aprendizaje de máquina. El algoritmo genético enfatiza la importancia de la cruce sexual (operador principal) sobre el de la mutación (operador secundario) y usa selección probabilística.

Los AG, usan un lenguaje prestado de la biología, donde un *individuo* es una posible solución al problema. Entonces, cada individuo tiene

- un *cromosoma o genotipo*, que es su representación codificada (comúnmente binaria). Cada cromosoma esta formado por un conjunto de *genes*, que son los bits individuales.
- un *fenotipo*, que es la forma decodificada de la solución y que esta definida por el usuario de forma externa al AG
- una *adecuación*, que es el valor de la función objetivo evaluada en dicha solución y es el parámetro a través del cual los individuos son seleccionados.

Un conjunto de posibles soluciones en un tiempo de computo determinado se denomina *población*, las cuales van cambiando la composición de sus genotipos mediante la aplicación de diferentes *operadores evolutivos*, como son:

- *Entrecruzamiento*: operación entre dos individuos, llamados *padres*, mediante la cual parte de su cromosoma es intercambiado para generar nuevos individuos llamados *hijos*,
- *Mutación*: operación sobre el genotipo de un individuo, mediante la cual uno o mas de sus genes son alterados,
- *Selección*: operación sobre una población, mediante la cual los individuos con mejores adecuaciones son seleccionados para pasar su información genética a la siguiente generación.

De esta forma un AG en general esta formado por los siguientes componentes:

- una **representación** genética de las posibles soluciones,
- un mecanismo para generar una **población inicial**,
- una o más funciones objetivo que dan lugar a los valores de **adecuación** de cada individuo y que serán el parámetro de selección.
- un conjunto de **operadores genéticos** que alteran y transmiten la información genética de una generación a la siguiente,
- un conjunto adecuado de **parámetros** como tamaño poblacional, tasas de mutación, tasas de entrecruzamiento, etc.

§3.4.2. Optimización de un función simple usando AG

Para entender la forma de trabajar de los AG, tomemos un ejemplo sencillo [58]. Supongamos que queremos encontrar el máximo de $f(x) = x \sin(10\pi x) + 1$, en el intervalo $[-1, 2]$, como se muestra en la Fig. 3.12. En esta caso la solución analítica es directa igualando a cero la derivada de la función objetivo

$$f'(x) = \sin(10\pi x) + 10\pi x \cos(10\pi x) = 0,$$

lo cual de la función trascendental

$$\tan(10\pi x) = -10\pi x.$$

Resolviendo gráfica o numéricamente, se obtienen los siguientes puntos estacionarios

$$x_i = \frac{2i + 1}{20} - \epsilon_i,$$

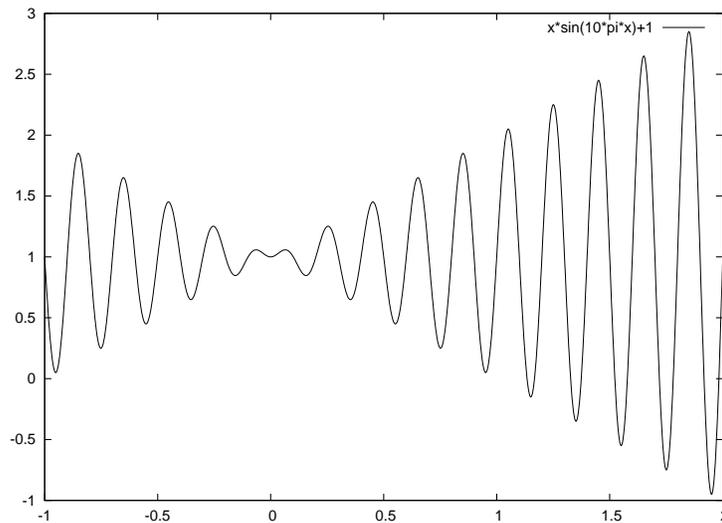


Figura 3.12: Gráfica de $f(x) = x \sin(10\pi x) + 1$; $x \in [-1, 2]$.

donde ϵ_i representa una secuencia decreciente de números reales que se aproximan a cero.

De esta forma tenemos por ejemplo que

$$x_{19} = \frac{37}{20} + \epsilon_{19} = 1.85 + \epsilon_i,$$

con lo que

$$f(x_{19}) \simeq f(1.85) = 2.85,$$

que corresponde al máximo buscado.

§3.4.3. Representación

Como se hace típicamente en AG, usaremos una representación binaria de los datos originales del problema. La longitud del genotipo, depende de la precisión requerida, que para este ejemplo fijaremos en seis lugares decimales.

Como el dominio de el problema tiene una longitud de tres, entonces para alcanzar la precisión requerida necesitamos dividir ese dominio en $3 \cdot 1000000$ partes iguales, lo cual implica un genotipo de 22 bits, dado que

$$2^{21} = 2097152 \leq 3000000 \leq 4194304 = 2^{22}.$$

El mapeo de una cadena binaria $\langle b_{21}b_{20} \dots b_0 \rangle$ a un número real $x \in [-1, 2]$ se logra mediante los siguientes pasos:

1. $(\langle b_{21}b_{20} \dots b_0 \rangle)_2 = \left(\sum_{i=0}^{21} b_i 2^i \right)_{10} = x'$
2. $x = -1 + x' \frac{3}{2^{22}-1}$

Así, el genotipo (1000101110110101000111) genera $x' = 2288967$ y por lo tanto se mapea a $x = -1 + 2288967 \frac{3}{4194303} = 0.637197$. Por su puesto los genotipos (0000000000000000000000) y el (1111111111111111111111) se mapean a los valores de frontera -1 y 2 respectivamente.

§3.4.4. Población inicial.

La creación de la población inicial suele hacerse de forma aleatoria, a menos que se tenga ya una población candidato a partir de la cual se quiera comenzar el proceso de optimización.

§3.4.5. Evaluación de la Adecuación

La función objetivo en este ejemplo es la mismo función $f(x)$ de modo que la selección se hará a partir de la función selectora

$$eval(v) = f(x),$$

donde v es el genotipo que representa a la solución x .

Como su nombre lo sugiere, la operación de selección emula a la selección natural y es en sí la calve del funcionamiento de los AG. Por ejemplo, tomemos tres cromosomas

$$\begin{aligned} v_1 &= (1000101110110101000111), \\ v_2 &= (0000001110000000010000), \\ v_3 &= (1110000000111111000101), \end{aligned}$$

que corresponden a los valores $x_1 = 0.637197$, $x_2 = -0.958973$ and $x_3 = 1.627888$, respectivamente. Evaluando la adecuación de cada uno, tenemos que

$$\begin{aligned} eval(v_1) &= 1.586345, \\ eval(v_2) &= 0.078878, \\ eval(v_3) &= 2.250650. \end{aligned}$$

Claramente el individuo con mayor adecuación y que debería ser seleccionado para transmitir su información genética a la siguiente generación sería el individuo x_3 .

§3.4.6. Operadores genéticos

Para este ejemplo se emplearan solo dos operadores clásicos: mutación y entrecruzamiento.

La mutación altera el bit de un gen particular con una probabilidad igual a la tasa de mutación. Supongamos que después de haber sido seleccionado el genotipo v_3 , se escoge de forma aleatoria su quinto gen para ser mutado, es decir que ese bit pasara de ser 0 a ser 1, con lo que el nuevo genotipo sera

$$(1110 \underbrace{0}_{\text{5to gen}} 00000111111000101) \longrightarrow (1110 \underbrace{1}_{\text{5to gen}} 00000111111000101)$$

Esta mutación evaluada da un valor de adecuación de -0.082257 lo cual representa una disminución significativa de su adecuación. Sin embargo, si en lugar del quinto gen, hubiera sido seleccionado el décimo, entonces el valor de adecuación mejoraría a un valor de 2.343555.

Ahora supongamos que después de haber sido seleccionados los genotipos v_3, v_2 para entrecruzarse, se escoge de forma aleatoria el quinto gen para iniciar la operación genética.

$$(00000|01110000000010000) \longrightarrow (00000|00000111111000101)$$

$$(11100|00000111111000101) \longrightarrow (11100|01110000000010000)$$

Esta recombinación particular, da valores de adecuación: 0.940865, 2.459245

Algorithm 3.1 Algoritmo general de un AG.

```

main
t ← 0
  incializa P(t)
  evalua P(t)
  while (no haya condición termino)
    do
      t ← t + 1
      selecciona P'(t) de P(t - 1)
      altera P(t)
      evalua P(t)
    end do
  end while
end main

```

§3.4.7. Parámetros

Para este problema, definamos los parámetros tamaño poblacional en 50 individuos, la tasa de entrecruzamiento en 0.25 y la de mutación en 0.01. Hecho esto, podemos implementar el Algoritmo 3.1 para obtener los resultados experimentales del siguiente cuadro.

Generación	Adecuación
1	1.441942
6	2.250003
8	2.250283
9	2.250284
10	2.250363
12	2.328077
39	2.344251
40	2.345087
51	2.738930
99	2.849246
137	2.850217
145	2.850227

§3.4.8. Algunas aplicaciones en agua subterránea

Calibración multiobjetivo.

En la práctica, un problema que surge con mucha frecuencia es la calibración de los modelos hidrogeológicos. Esta calibración suele hacerse mediante la búsqueda iterativa de parámetros que hagan que el modelo reproduzca de mejor manera los resultados de campo. Para hacer este proceso más eficiente, se han desarrollado métodos tanto locales [62, 2], como globales [32, 73]. Sin embargo, todos ellos se basan en la minimización de una sola función objetivo, típicamente el error cuadrático medio entre resultados de simulación y datos de campo. En muchas aplicaciones por el contrario al caso anterior, es deseable que el modelo reproduzca diferentes tipos de observaciones, por lo cual una calibración multiobjetivo es necesaria.

La manera en como se ha venido resolviendo el problema de la calibración multiobjetivo, es mediante el uso de una sola función objetivo, construida mediante la combinación lineal de los diferentes tipos de datos, asignándole a cada uno un peso diferente [45]. Por supuesto, estos métodos tienen el gran problema de que los valores para los pesos se escogen de manera arbitraria o “a criterio de experto”, por lo cual se desarrollaron esquemas en los cuales se van variando los pesos de forma sistemática.

Una alternativa a esto es la introducción de AG capaces de resolver el problema multiobjetivo de forma simultánea, evitando así el uso de pesos y obteniendo un mayor entendimiento del modelo, al estudiar la forma en como los diferentes objetivos compiten entre si, llegando incluso a revelar si algún proceso físico importante no fue considerado en el modelo conceptual [64].

Un ejemplo de este tipo de calibración [67] se aplico a un modelo de agua superficial-subterránea en el valle Yaki de Sonora, México. Esta es una región importante de cultivo por riego de aproximadamente 6,800 km, ubicada junto al mar de Cortéz.

Se llevo a cabo una calibración multiobjetivo usando considerando las cargas hidráulicas para los acuíferos someros, para acuíferos profundos; así como volúmenes de agua infiltrada de los drenajes y canales agrícolas.

La mayor ventaja de haber usado la aproximación de los AG en este estudio es que se pudo calibrar simultáneamente tanto los parámetros como las incertidumbres estructurales del modelo. En este caso, los resultados de la calibración hicieron evidente que era necesario incluir en el modelo conceptual el efecto de tener el suelo en barbecho durante ciertas épocas del año.

Capítulo 4

SMORN y su aplicación en agua subterránea.

La motivación principal para generar una aplicación de gestión óptima en recursos naturales es la necesidad, reconocida nacional e internacionalmente, de hacer un uso sustentable de los mismos. En particular hemos mostrado la importancia del agua subterránea tanto a nivel local como global; desde un punto de vista del aprovechamiento directo del recurso; en el aseguramiento del suministro de alimentos; pero también desde la perspectiva de desarrollo social y económico que su uso genera. Sin embargo, mostramos también que el problema de flujo subterráneo al ser modelado por EDP de 2° orden es difícil de optimizar y sus problemas de gestión no solo deben tomar en cuenta aspectos hidrogeológicos, sino también económicos y sociales, por lo que dan lugar de forma natural a problemas multiobjetivos de difícil resolución. Por lo tanto nuevas metodologías de gestión son necesarias para poder dar respuesta a los los problemas que en agua subterránea están surgiendo actualmente

§4.1. Objetivos

1. Generar una metodología de control óptimo para recursos naturales que sea:
 - a)* Modular: que cada parte del proceso pueda ser tratado como un modulo individual, de tal forma que futuras modificaciones en un módulo no comprometan la funcionalidad del sistema.
 - b)* Flexible: que se pueda aplicar tanto a diferentes recursos naturales como a distintas clases de problemas de gestión.
 - c)* Orientada al usuario final: que demande el menor conocimiento del especialista por parte del usuario final.



Figura 4.1: Estructura general de SMORN

2. Implementar esta metodología en un sistema computacional (SMORN) que sea:

- a) Libre: que todos los códigos usados sean libres o cuando menos abiertos.
- b) Económico: que desde un punto de vista computacional, demande un mínimo de recursos físicos.
- c) Robusto: que el mismo código pueda ser usado para resolver muchos tipos de problema diferentes.

3. Aplicar la implementación computacional a un problema real de gestión en agua subterránea, para mostrar su viabilidad como herramienta de creación de políticas y toma de decisiones.

§4.2. Estructura de SMORN

El Sistema de Manejo Óptimo de Recursos Naturales (SMORN), es un método de gestión que en principio podría aplicarse al manejo óptimo de cualquier recurso natural, del cual se pueda establecer un modelo matemático y computacional.

A partir de la modelación del recurso se establecen las funciones objetivo, las cuales son optimizadas mediante computo evolutivo. En particular, SMORN usa una versión modificada por mi de *Single and Multiobjective Genetic Algorithm Toolbox in C++ (GAtoolbox)* por Kumara Sastry del grupo de algoritmos genéticos (IlliGAL) de la universidad de Illinois, Estados Unidos. Posterior al proceso de optimización, que genera un conjunto de posibles soluciones óptimas equivalentes, conocidas como frente de Pareto, SMORN incorpora el uso de la teoría de juegos para analizar los resultados obtenidos desde una perspectiva ya de toma de decisiones. Una solución \tilde{X} es un óptimo de Pareto si no existe un vector X que haga mejorar alguno de los objetivos, sin que empeore de forma simultánea alguno de los otros. En general, la solución en el sentido de Pareto al problema de optimización multiobjetivo no será única: la solución estará formada por el conjunto de todos los óptimos de Pareto, conjunto que se conoce como frente de Pareto.

Dada la estructura general de SMORN como se muestra en la Fig.4.1, se puede apreciar que este es un sistema absolutamente interdisciplinario que involucra tres etapas básicas:

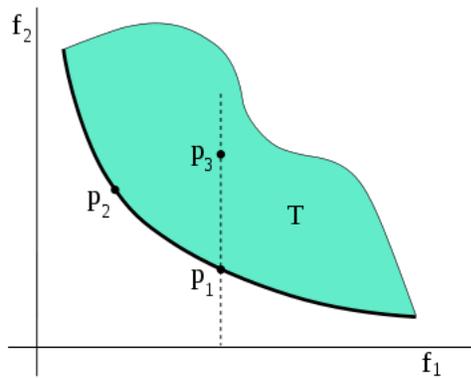
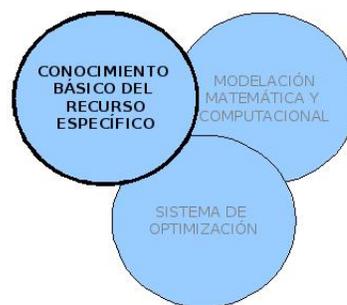


Figura 4.2: Siendo cada eje de esta gráfica un objetivo del problema multiobjetivo (f_1, f_2) y el área coloreada T la imagen de dicho problema, entonces los puntos P_1 y P_2 son dos soluciones de Pareto. La línea gruesa representa entonces el frente de Pareto asociado al problema, de tal forma que el punto P_2 no podría mejorar su valor para la función objetivo f_1 a menos que disminuya al mismo tiempo su correspondiente valor de f_2 . Finalmente se puede ver que ningún punto P_3 dentro de la zona T puede tener un mejor juego de valores para las funciones objetivo que los obtenidos sobre el frente de Pareto.

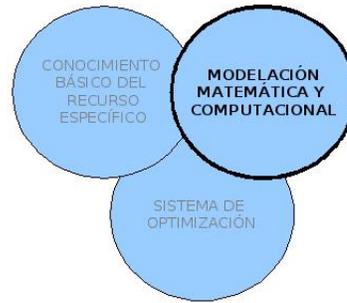
■ Conocimiento Básico.

- Hidrogeología
- Edafología
- Física
- Química
- Ecología
- Economía
- Sociología



■ Modelación

- EDO, EDP
- Minería de Datos, Autómatas Celulares, otros
- Métodos Numéricos
- Geoestadística, SIG's
- fortran, C++, awk-sed, peral, scilba, octave, modelo



■ Optimización

- Control clásico
- Control óptimo
- Programación no lineal
- Programación dinámica
- Inteligencia Artificial: AG, computo evolutivo, sistemas expertos



§4.2.1. El código de optimización.

El motor de optimización de SMORN es el GAtoolbox, el cual es un software libre¹ para aplicaciones académicas, que permite usar diferentes algoritmos de selección, recombinación, mutación, nicho y manejo de constricciones. Así mismo, el GAtoolbox puede resolver problemas de optimización de una función objetivo sujetas a n constricciones o bien un problema de m funciones objetivo con n constricciones. Finalmente, el código es altamente modular al estar escrito en C++, por lo cual se pueden incorporar otros operadores, así como adecuarlo a problemas particulares. Para este fin, basta escribir el archivo `userDefinables.cpp` correspondiente al problema de optimización particular que se quiera resolver.

Acceso al código

El código original se puede bajar de <ftp://ftp-illigal.ge.uiuc.edu/pub/src/GA/GAtoolbox.tgz>. Como esta tesis esta hecha completamente con software libre, las instruc-

¹libre en el sentido de la Fundación de Software Libre (www.fsf.org)

ciones de uso corresponden a un sistema Gnu-Linux, en particular Debian.

Una vez que se ha bajado el archivo GAtoolbox.tgz, abrir una terminal y escribir el siguiente código

```
$tar zxvf GAtoolbox.tgz
```

lo cual debe generar un directorio con los siguientes archivos:

DISCLAIMER	README	Makefile	chromosome.cpp
chromosome.hpp	crossover.cpp	crossover.hpp	ga.cpp
ga.hpp	globalSetup.cpp	globalSetup.hpp	individual.cpp
individual.hpp	localsearch.cpp	localsearch.hpp	nsgapopulation.cpp
population.cpp	population.hpp	random.cpp	random.hpp
selection.cpp	selection.hpp	userDifinables.cpp	

Compilar y correr el código.

Para compilar el código, es necesario tener instalado los compilador de C y C++, yo use gcc 4.3 y gobjc++ 4.3. Si no hay ninguna librería faltante, en cuyo caso deberá instalarse, basta hacer

```
$make
```

en el directorio donde se descomprimió, lo cual generará un archivo ejecutable llamado GAtbx

El ejecutable GAtbx usa un parámetro, que es el archivo de configuración, en donde se declaran entre otras cosas si el problema es de una función objetivo o multiobjetivo, numero de constricciones, tamaño de la población, etc. El software viene con dos ejemplos de problemas; el primero es un problema de una función objetivo con dos constricciones, mientras el segundo es un problema de dos funciones objetivo. Asimismo, se proveen cuatro tipos de archivos de configuración: `input sga maxSpec`, `input sga minSpec`, `input nsga maxSpec`, y `input nsga minSpec`.

El ejemplo de una función objetivo resuelve el problema de la función de Himmelblau [Rekalaitis, Ravindran, & Ragsdell, 1983] dada por

$$f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2,$$

sujeta a las constricciones

$$(x_1 - 5)^2 + x_2^2 \leq 26,$$

$$4x_1 + x_2 \leq 20.$$

El caso multiobjetivo, resuelve el problema de Kurosawe [Deb, Pratap, Agrawal, & Meyarivan, 2002)], dado por

$$f_1(x_1, x_2, x_3) = \sum_{i=1}^2 \left[-10 \exp \left(-0.2 \sqrt{x_i^2 + x_{i+1}^2} \right) \right],$$

$$f_2(x_1, x_2, x_3) = \sum_{i=1}^3 [|x_i|^{0.8} + 5 \sin x_i^3].$$

En general, para correr el GAtoolbox, se debe escribir desde la terminal

```
$GAtbx input_file_name
```

Al correrlo, se mostrarán en pantalla un serie de datos estadísticos interesantes, dependiendo del tipo de problema que se esta resolviendo, una o varias funciones objetivo.

Capacidades del AG

El núcleo del GAtoolbox es el Algoritmo Evolutivo para Optimización Multiobjetivo NSGA-II [Deb, Pratap, Agrawal, & Meyarivan, 2002]. Una ventaja de este código es que la codificación de las variables de decisión se puede hacer con reales o enteros, especificando sus respectivos intervalos. Este procedimiento de codificación permite que las variables de decisión sean, binarias, discretas, arreglos, alfanuméricas o números reales. El tipo de codificación debe ser especificado en el archivo de configuración, siendo las opciones `double` o `int`.

Los operadores genéticos que están actualmente implementados son:

1. Selección

- a) Torneo con reemplazo. [Goldberg, Korb, & Deb, 1989; Sastry & Goldberg, 2001]
- b) Torneo sin reemplazo [Goldberg, Korb, & Deb, 1989; Sastry & Goldberg, 2001]
- c) Selección truncada [Mühlenbein & Schlierkamp-Voosen, 1993]
- d) Ruleta [Goldberg, 1989]
- e) Selección Universal Estocástica (SUS) [Baker, 1985; Baker, 1987; Grefenstette & Baker, 1989; Goldberg, 1989]

2. Recombinación / Crossover

- a) Un punto de recombinación [Goldberg, 1989; Sastry, Goldberg, & Kendall, 2005]
- b) Dos puntos de recombinación [Goldberg, 1989; Sastry, Goldberg, & Kendall, 2005]

- c)* Recombinación uniforme [Goldberg, 1989; Sastry, Goldberg, & Kendall, 2005]
- d)* Recombinación Binaria Simulada (SBX) [Deb & Agarwal, 1995; Deb & Kumar, 1995]

3. Mutación

- a)* Mutación Selectiva, donde un gen escogido aleatoriamente es reemplazado por otro tomado con una probabilidad p_m de una distribución de probabilidad uniforme con intervalos $[x_{i,min}, x_{i,max}]$.
- b)* Mutación gaussiana (Genewise), donde todos los genes pueden ser mutados con una probabilidad p_m , usando una distribución de probabilidad Gaussiana.
- c)* Mutación polinomial [Deb & Agarwal, 1995; Deb & Kumar, 1995; Deb, 2001]

4. Niching

- a)* Compartir adecuación [Goldberg & Richardson, 1987; Goldberg, 1989]
- b)* Atracción determinística [Mahfoud, 1992]
- c)* Torneo de selección restringida [Harik, 1995]

5. Escalamiento

- a)* Clasificación lineal [Baker, 1985; Goldberg, 1989]
- b)* Escalamiento Sigma [Forrest, 1985; Goldberg, 1989]
- c)* El usado por default, que solo es usado si alguna de las siguientes condiciones se cumple: i) el proceso de selección es de ruleta o SUS y la adecuación mínima es negativa, o cuando ii) el método de niching es compartir la adecuación y la adecuación mínima es negativa.

6. Manejo de constricciones

- a)* Método de torneo [Deb, 2001]
- b)* Método lineal de penalización, donde una suma pesada se lleva a cabo sobre las violaciones de un individuo, de tal forma que una fracción de la penalización máxima es sumada o restada de las funciones objetivos, dependiendo si el problema era de minimización o maximización.

- c) Método de penalización cuadrática, similar al lineal, con la excepción de que la suma se hace sobre el cuadrado de las violaciones a las restricciones.²
7. Búsqueda Local. El método empleado para la búsqueda local es el de Nelder-Mead [Nelder & Mead, 1965; Press, Flannery, Teukolsky, & Vettering, 1989]. La solución obtenida por la búsqueda local es incorporada usando
- a) Estrategia de Baldwin, donde reemplazan a individuos encontrados normalmente, pero se mantiene su adecuación.
 - b) Estrategia Lamarkiana, donde tanto el individuo como su adecuación son sustituidos por los obtenidos por la búsqueda local.
8. Reemplazo elitista. En general el GAtoolbox, los individuos se escogen según su adecuación y valores de violación de constricciones. Con el reemplazo elitista, una proporción p_e de individuos altamente adecuados (llamados de elite) es retenido, mientras que el restante $n(1 - p_e)$ es reemplazado por los individuos elite.

Criterios de término.

El GAtoolbox provee diferentes criterios de término:

1. Número de evaluaciones de las funciones objetivo.
2. Varianza de la adecuación.
3. Mejor adecuación.
4. Adecuación promedio.
5. Promedio de las funciones objetivo.
6. Cambio en la adecuación del mejor individuo.
7. Cambio en la varianza de la adecuación.
8. Cambio en la mejor evaluación de las funciones objetivos.
9. Cambio en el promedio de las funciones objetivo.
10. Número de frentes de pareto.

²Hay que hacer notar que la forma en que se manejan las restricciones usa el valor de la violación y no el valor de la función de constricción, por ejemplo, si la constricción es $10x_1 + x_2 \leq 20$ de tal forma que si $x_1 = 2$, $y x_2 = 2$, entonces el valor que corresponde es 2. Por otro lado si $x_1 = 1$ $y x_2 = 7$, entonces el valor será 0.

11. Número de individuos en el primer frente de pareto.
12. Cambio en el numero de frentes de pareto.

Archivo de configuración.

En el archivo de configuración, todas las líneas en blanco son ignoradas, así como las que comienzan con el símbolo #, que son consideradas comentarios. Todos los parámetros son leídos en un orden estricto, por lo que no debe bajo ningún motivo cambiarse el orden de las líneas de configuración. Para beneficio del usuario, el archivo de configuración viene comentado a detalle. Como un ejemplo mostraremos el archivo input `sga maxSpec`, incluyendo los números de las líneas de código.

```
1 #
2 # GA type: SGA or NSGA
3 # 4 SGA
4
5
6 #
7 # Number of decision variables
8 #
9 2
10
11 #
12 # For each decision variable, enter:
13 # decision variable type, Lower bound, Upper bound
14 # Decision variable type can be double or int
15 #
16 double 0.0 6.0
17 double 0.0 6.0
18
19 #
20 # Objectives:
21 # Number of objectives
22 # For each objective enter the optimization type: Max or Min
23 #
24 1
25 Min
26
27 #
28 # Constraints:
29 # Number of constraints
30 # For each constraint enter a penalty weight
31 #
32 2
33 1.0
34 1.0
35 #
36 # General parameters: If these parameters are not entered default
37 # values will be chosen. However you must enter
38 # "default" in the place of the parameter.
39 # [population size]
40 # [maximum generations]
```

```
42 # [replace proportion]
43 #
44 100
45 100
46 0.9
47
48 #
49 # Niching (for maintaining multiple solutions)
50 # To use default setting type "default"
51 # Usage: Niching type, [parameter(s)...]
52 # Valid Niching types and optional parameters are:
53 # NoNiching
7
54 # Sharing [niching radius] [scaling factor]
55 # RTS [Window size]
56 # DeterministicCrowding
57 #
58 # When using NSGA, it must be NoNiching (OFF).
59 #
60 NoNiching
61
62 #
63 # Selection
64 # Usage: Selection type, [parameter(s)...]
65 # To use the default setting type "default"
66 #
67 # Valid selection types and optional parameters are:
68 # RouletteWheel
69 # SUS
70 # TournamentWOR [tournament size]
71 # TournamentWR [tournament size]
72 # Truncation [# copies]
73 #
74 # When using NSGA, it can be neither SUS nor RouletteWheel.
75 #
76 TournamentWOR 2
77
78 #
79 # Crossover
80 # Crossover probability
81 # To use the default setting type "default"
82 #
83 # Usage: Crossover type, [parameter(s)...]
84 # To use the default crossover method type "default"
85 # Valid crossover types and optional parameters are
86 # OnePoint
87 # TwoPoint
88 # Uniform [genewise swap probability]
89 # SBX [genewise swap probability][order of the polynomial]
90 #
91 0.9
92 SBX 0.5 10
93
94 #
95 # Mutation
```

```
96 # Mutation probability
97 # To use the default setting type "default"
98 #
99 # Usage: Mutation type, [parameter(s)...]
100 # Valid mutation types and the optional parameters are:
101 # Selective
8
102 # Polynomial [order of the polynomial]
103 # Genewise [sigma for gene #1][sigma for gene #2]...[sigma for gene #ell]
104 #
105 0.1
106 Polynomial 20
107
108 #
109 # Scaling method
110 # To use the default setting type "default"
111 #
112 # Usage: Scaling method, [parameter(s)...]
113 # Valid scaling methods and optional parameters are:
114 # NoScaling
115 # Ranking
116 # SigmaScaling [scaling parameter]
117 #
118 NoScaling
119
120 #
121 # Constraint-handling method
122 # To use the default setting type "default"
123 #
124 # Usage: Constraint handling method, [parameters(s)...]
125 # Valid constraint handling methods and optional parameters are
126 # NoConstraints
127 # Tournament
128 # Penalty [Linear|Quadratic]
129 #
130 Tournament
131
132 #
133 # Local search method
134 # To use the default setting type "default"
135 #
136 # Usage: localSearchMethod, [maxLocalTolerance], [maxLocalEvaluations],
137 # [initialLocalPenaltyParameter], [localUpdateParameter],
138 # [lamarckianProbability], [localSearchProbability]
139 #
140 # Valid local search methods are: NoLocalSearch and SimplexSearch
141 #
142 # For example, SimplexSearch 0.001000 20 0.500000 2.000000 0.000000 0.000000
143 NoLocalSearch
144
145 #
146 # Stopping criteria
147 # To use the default setting type "default"
148 #
149 # Number of stopping criterias
```

```
9
150 #
151 # If the number is greater than zero
152 # Number of generation window
153 # Stopping criterion, Criterion parameter
154 #
155 # Valid stopping criterias and the associated parameters are
156 # NoOfEvaluations, Maximum number of function evaluations
157 # FitnessVariance, Minimum fitness variance
158 # AverageFitness, Maximum value
159 # AverageObjective, Max/Min value
160 # ChangeInBestFitness, Minimum change
161 # ChangeInAvgFitness, Minimum change
162 # ChangeInFitnessVar, Minimum change
163 # ChangeInBestObjective, Minimum change
164 # ChangeInAvgObjective, Minimum change
165 # NoOfFronts (NSGA only), Minimum number
166 # NoOfGuysInFirstFront (NSGA only), Minimum number
167 # ChangeInNoOfFronts (NSGA only), Minimum change
168 # BestFitness (SGA with NoNiching only), Maximum value
169 #
170 0
171
172 #
173 # Load the initial population from a file or not
174 # To use the default setting type "default"
175 #
176 # Usage: Load population (0|1)
177 #
178 # For example, if you want random initialization type 0
179 # On the other and if you want to load the initial population from a
180 # file, type
181 # 1 <population file name> [0|1]
182 #
183 # Valid options for "Load population" are 0/1
184 # If you type "1" you must specify the name of the file to load the
185 # population from. The second optional parameter which indicates
186 # whether to evaluate the individuals of the loaded population or not.
187 0
188
189 # Save the evaluated individuals to a file
190 #
191 # To use default setting type "default".
192 #
193 # Here by default all evaluated individuals are stored and you will be
194 # asked for a file name later when you run the executable.
195 #
196 # Usage: Save population (0|1)
197 # For example, if you don't want to save the evaluated solutions type 0
198 # On the other and if you want to save the evaluated solutions
199 # 1 <save file name>
200 #
201 # Note that the evaluated solutions will be appended to the file.
202 #
```

```
203 # Valid options for "Save population" are 0/1
204 # If you type "1" you must specify the name of the file to save the
205 # population to.
206 1 evaluatedSolutions.txt
207
208 #END
```

Los archivos

La implementación del GAtoolbox no hace uso de las opciones avanzadas de C++ como pueden ser los templates o las herencias. Esta es una gran ventaja, pues implica que uno no tiene necesariamente que ser un experto en C++ para poder modificar el código. De hecho uno podría modificar el código e implementar su propia aplicación usando enteramente comandos de C.

Cada archivo *.cpp tiene su correspondiente *.hpp, que son los encabezados y contienen las declaraciones de varias clases, a excepción de nsgapopulation.cpp y userDefinalbes.cpp.

Si uno no desea introducir o modificar los operadores genéticos que vienen junto con el código original, el único archivo que hay que editar es el de userDefinables.cpp.

En la Fig. §4.2.1 se muestra el diagrama de flujo del motor de optimización de SMORN.

§4.3. Sitio de estudio.

El más largo de los ríos interiores del país, el Lerma, tiene su origen en los manantiales de Almoloya del Río, en el estado de México; atraviesa hacia el noroeste del Valle de Toluca, y desagua en el lago de Chapala. Tiene una longitud de 708 km y drena una cuenca de 47.116 km².

La importancia del Río Lerma estriba en que es un recurso natural para las actividades humanas básicas de la región, en que sirve como fuente de eléctrica para la ciudad de México con 79.000 Kw, además que provee agua potable por un acueducto que atraviesa la Sierra de las Cruces (García y Falcón, 1974).

Por otra parte, se utiliza para riego de los cultivos de las zonas agrícolas situadas en sus márgenes. Como ejemplo, tenemos el Valle de Santiago en el Estado de Guanajuato, el cual ocupa un lugar importante dentro de las zonas agrícolas del país.

Pero antes de verter sus aguas en el lago de Chapala, el río Lerma recibe los escurrimientos de uno de sus principales afluentes: el río Duero. En su trayectoria desde sus manantiales más altos, el Duero humedece las fértiles llanuras de milenarios suelos de aluvión de la Cañada de los Once Pueblos y los valles de Zamora y la Ciénega de Chapala. La afortunada conjunción de tierras fértiles y agua dieron fama a la región por sus abundantes cosechas y por la riqueza

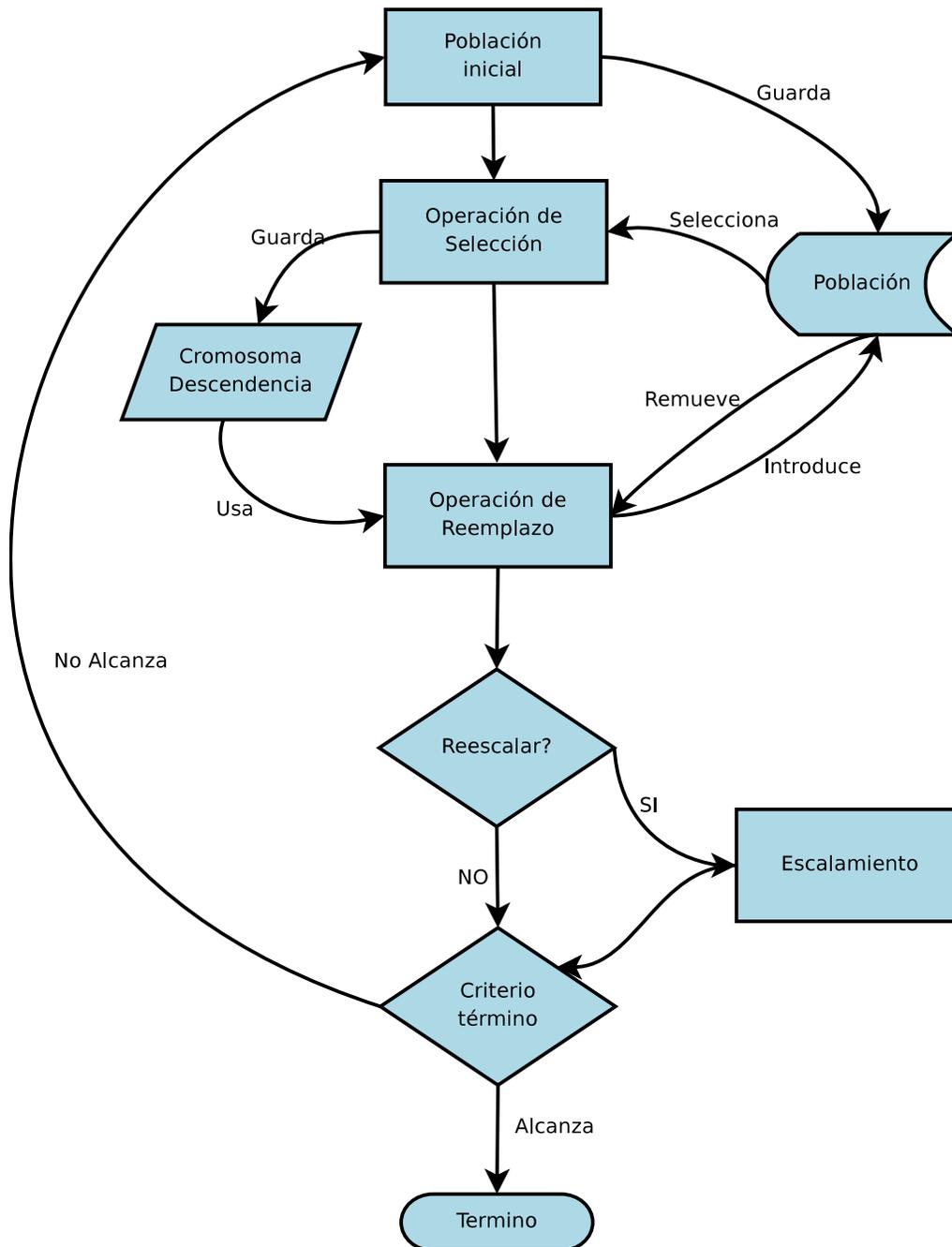


Figura 4.3: Diagrama de flujo del motor de optimización de SMORN, el AGtoolbox.

y prosperidad de sus habitantes desde épocas muy remotas hasta la actualidad. Proceden de aquí la fresa, la papa, diversas hortalizas como el brócoli, la cebolla, el jitomate, el chile, el pepino, la calabacita y la calabaza, que llegan a los mercados nacionales y de países extranjeros, en tanto que el maíz, el trigo, la garbanza y el garbanzo alimentan a poblaciones cercanas y lejanas de manera directa o mediada por procesos industriales.

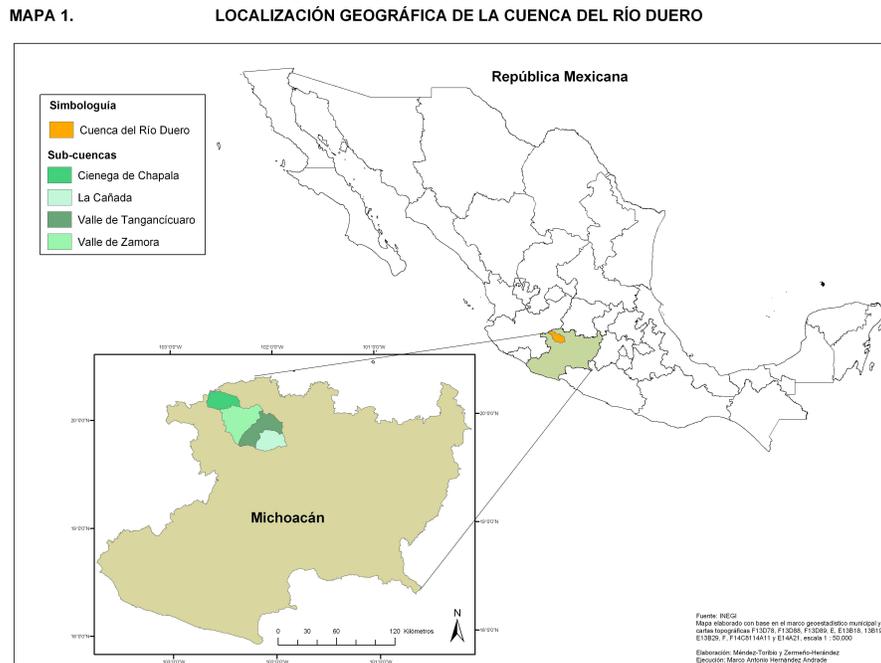


Figura 4.4: Localización de la cuenca del río Duero

§4.3.1. Hidrología de la cuenca.

La cuenca del río Duero se encuentra delimitada por las coordenadas $19^{\circ}40'$ y $20^{\circ}20'$ N y $101^{\circ}52'$ a $102^{\circ}40'$ O. Con una altitud media de 2000 m.s.n.m. el río Duero es parte del sistema hidrológico Lerma-Chapala-Santiago, que a su vez es el más importante del Altiplano Central de México.

La cuenca cubre una superficie de 3512 km^2 , con un escurrimiento promedio anual de 250 millones de metros cúbicos. Corriendo en dirección SE-NO, cruza la cañada de los Once Pueblos, los valles de Tangancicuaro (o Guadalupe) y Zamora para, finalmente, fluir hacia la Ciénega de Chapala.

El río tiene su origen en el Cerro de la Loma como escurrimiento varios manantiales recibiendo el nombre de río Chilchota. Después del poblado de Chilchota y antes de ingresar al valle de Tangancicuaro (Guadalupe), el caudal del río es dividido en varios canales no

revestidos, que reciben aportes de manantiales de agua fría. La siguiente zona de tránsito es el valle de Guadalupe (Tangancícuaro), donde sus principales aportes son el arroyo El Pejo, por su margen izquierdo, poco antes de llegar a Tangancícuaro, y el río Tlazazalca, por la margen derecha, con aportes de la presa Urepetiro.

Después de la confluencia del río Tlazazalca se encuentra la estación hidrométrica Camecuaró; punto en el que el área drenada es de aproximadamente 1221 km^2 . Luego, por su margen izquierdo, el río recibe un caudal medio de $2 \text{ m}^3/\text{s}$ siendo el afluente más importante del canal de salida del lago Camecuaró.

En la subcuenca del Valle de Zamora, delimitada al norte por los cerros de La Estanzuela y Las Chivas; el río es parte principal del Distrito de Riego número 061, el más grande del estado de Michoacán. Sus aguas son utilizadas para alimentar a la hidroeléctrica “El Platanal”, así como para irrigar los numerosos campos de cultivo. Su principal aporte proviene del río Celio además de recibir los desechos de la ciudad de Zamora y Jacona.

Finalmente el Río Duero termina su recorrido en la Ciénaga de Chapala. En toda esta zona el río ha sido modificado y canalizado.

§4.3.2. Hidrogeología

Centrando la atención en el valle de Zamora, podemos hacer una descripción geohidrológica general de la cuenca. En las sierras que rodean al valle afloran basaltos del Cuaternario, subyaciendo a rocas volcánicas de menor permeabilidad. Por su textura vesicular y denso fracturamiento, tienen permeabilidad y capacidad de infiltración muy altas, por tanto, son excelentes receptores de recarga y constituyen acuíferos muy permeables en el subsuelo de la porción sur del valle de Zamora, donde están cubiertos por depósitos aluviales del Reciente. Por el contrario, los basaltos del Terciario son de capacidad de infiltración y permeabilidad muy bajos, debido a que su estructura es masiva y poco fracturada. En el subsuelo de la cuenca presenta una configuración escalonada, a causa del tectonismo, y constituyen el basamento geohidrológico del acuífero. El acuífero principal está formado por derrames basálticos, piroclásticos, materiales aluviales y depósitos lacustres. Los tres primeros son los elementos más permeables, y los últimos constituyen acuitardos de permeabilidad media a baja. Su espesor aumenta de los bordes hacia el centro del valle, donde alcanza valores hasta de 350 m; está limitado inferiormente por depósitos lacustres antiguos, y en las porciones bajas del valle está semiconfinado por los depósitos lacustres recientes.

Los coeficientes de transmisividad obtenidos, varían de 0.000036 y $0.044 \text{ m}^2/\text{s}$; los valores más altos -0.009 a $0.044 \text{ m}^2/\text{s}$ corresponden al Valle de Tangancícuaro y están asociados con materiales granulares y basaltos; los más bajos corresponden al Valle de Zamora -0.000036 a $0.0024 \text{ m}^2/\text{s}$, donde el relleno acuífero es de menor permeabilidad. Los valores iniciales

Uso	# Pozos	%	Vol. Extraído [$10^6 m^3/año$]
Agropecuario	382	51.5	74.231
Urbano	161	21.7	22.884
Industrial	77	10.4	2.477
Otros	122	16.4	7.552
Sumas	742	100	107.144

Cuadro 4.1: Censo de aprovechamiento de agua subterránea 1977

del coeficiente de almacenamiento se fijaron a partir de consideraciones sobre la litología y el tipo de acuífero, ya que no pudieron ser obtenidos por medio de pruebas de bombeo. Como el acuífero que nos ocupa es del tipo “libre”, el coeficiente de almacenamiento es igual, numéricamente, al rendimiento específico, siendo este de 0.005 a 0.3, dependiendo de su granulometría; sin embargo, como en el área estudiada existen materiales arcillosos, se estimó un valor representativo del rendimiento específico no mayor de 0.10.

El último censo que se realizó en esta zona fue en 1977, censándose un total de 250 captaciones de aguas subterráneas, a la fecha en los registros, en los cuales existen alrededor de 742 aprovechamientos de agua subterránea, los cuales tienen la siguiente distribución por tipo de uso:

§4.3.3. Geología

Localizada en su mayoría dentro del dominio de las cuencas lacustres, la cuenca del río Duero, se caracterizan por depósitos superficiales (aluviones y lacustres y lacustres recientes). Hacia el sur existe otro dominio importante dentro del cinturón volcánico Transmexicano caracterizado por la presencia de conos de escorias y flujos piroclásticos (Garduño et al., 2003).

En La Cañada de los Once Pueblos así como en los valles agrícolas se han acumulado sedimentos aluviales a lo largo del Cuaternario, que fueron conformando suelo areno-arcilloso, producto de la erosión de las rocas ígneas, de tamaño variable, de formas subángulosas y subredondeadas. Estos depósitos llegan a tener espesores de cientos de metros en la Ciénega de Chapala, antigua planicie de inundación de los ríos Lerma y Duero.

§4.3.4. Suelos y Clima

Los suelos del municipio datan del periodo cenozoico terciario inferior, entre los que predominan: Andosol, Cambisol, Luvisol (carta edafológica INEGI), con los siguientes porcentajes estimados andosol 70 %, luvisol 20 %, cambisol 10 %. Andosol 70 %.

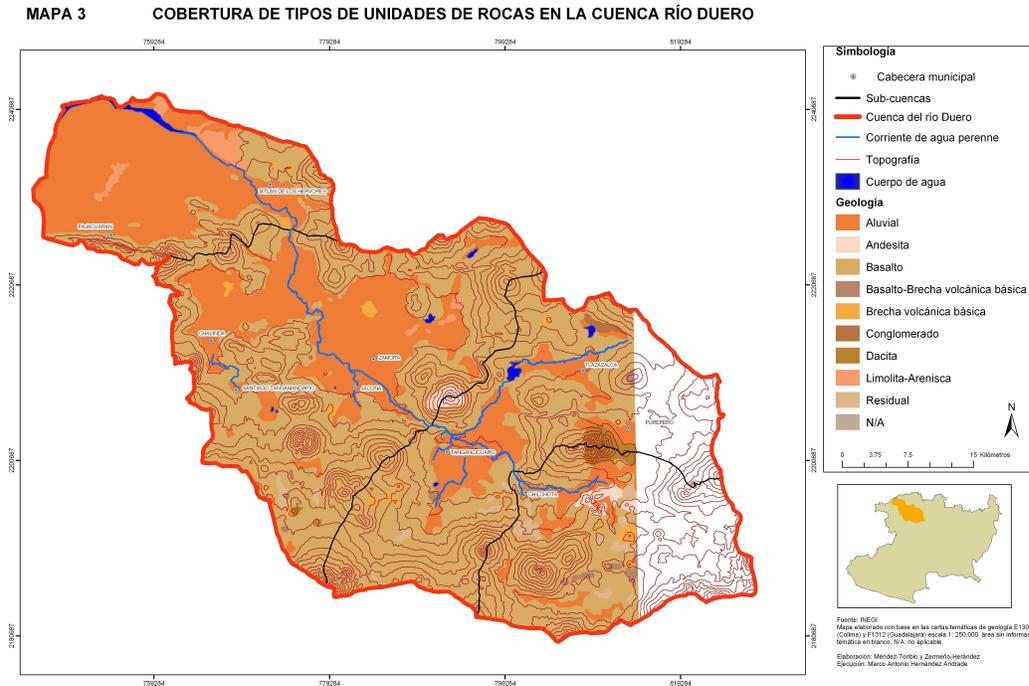


Figura 4.5: Mapa geológico de la cuenca del río Duero.

De acuerdo con la clasificación climática de Köppen, modificado por García (1988), la cuenca del río Duero presenta en general dos tipos climáticos A y C. La parte más elevada de la cuenca (2500 a 3000 msnm) en su porción sureste, presenta los subtipos Cw y Ce. La cuenca media y baja presenta un clima semicálido subhúmedo con lluvias en verano. El porcentaje de lluvias en invierno es menor al 55%, la temperatura media anual mínima es de 9.3° C y la máxima es de 25.3° C. Los meses más fríos son diciembre, enero y febrero, y la temporada más cálida abarca el periodo de marzo a julio. La precipitación anual oscila entre los 777 y 1146 mm. La evaporación anual varía entre los 1046 y 2019 mm anuales.

Como se mencionó en la sección anterior, el manejo de la cuenca del río Duero es importante desde varias perspectivas:

- Es parte del sistema hidrológico más importante del altiplano central.
- El recurso hídrico en la zona es usado para sostener una de las regiones agrícolas más importantes del país, así como para dotar de agua potable a la ciudad de Zamora, cuya densidad poblacional es considerable.
- Se espera que la extracción se incremente considerablemente en las próximas décadas debido al desarrollo regional.

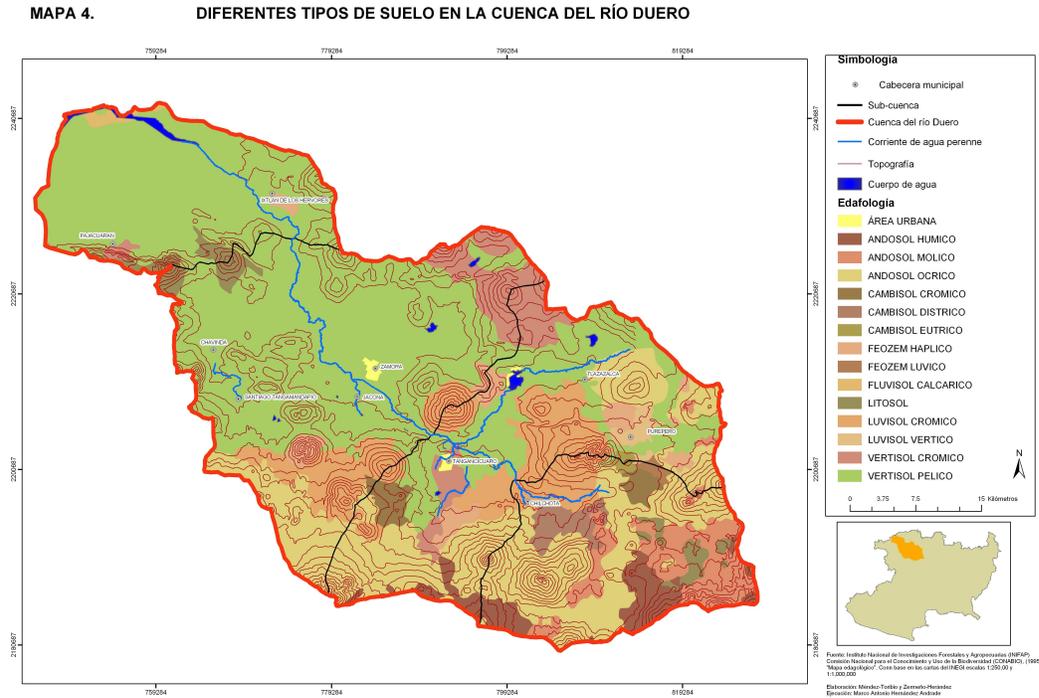


Figura 4.6: Mapa de suelos de la cuenca del río Duero.

- El manejo que se haga de la cuenca afecta directa o indirectamente el estado del lago de chapala, el embalse natural más importante del país.

Esto nos hace pensar inmediatamente en que una aproximación multiobjetivo es necesaria para poder diseñar un plan de manejo óptimo de la cuenca, por lo cual el uso de un sistema como SMORN suena natural.

§4.4. Modelo de la cuenca del Río Duero.

El modelo computacional del río Duero fue realizado por el M. en C. Felipe Armas, como parte de su tesis de grado, dirigida por el Dr. Oscar Escolero del Instituto de Geología de la UNAM. El modelo está implementado en MODFLOW 2000, que es un software de código abierto de la USGS que se puede descargar gratuitamente en <http://water.usgs.gov/nrp/gwsoftware/modflow2000/modflow2000.html>.

El problema de control que se planteó fue un problema multiobjetivo, con las siguientes funciones:

1. Maximizar el total de agua extraída del acuífero.
2. Minimizar el abatimiento promedio.

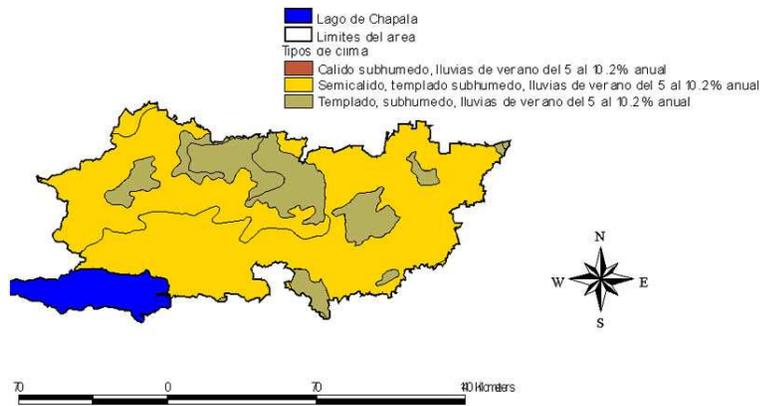


Figura 4.7: Mapa de Clima en la cuenca del río Duero [fao]

3. Minimizar la velocidad promedio de abatimiento.

La primera función objetivo es claramente una función diseñada para obtener el mayor beneficio del recurso, mientras que la segunda y tercera, además de buscar minimizar el impacto al acuífero tienen como intención disminuir posibles problemas de subsidencia.

Para resolver este problema del tipo NSGA en el GAtoolbox; se usaron 546 variables de decisión de tipo double, que corresponden a los posibles valores de extracción en cada pozo. El tamaño poblacional se estableció en 10 individuos, la proporción de reemplazo en 0.9 y el número máximo de generaciones en 2000. Para ver los demás parámetros de la optimización, favor de ver el apéndice.

En la Fig. §4.4 se muestra el diagrama de flujo de la implementación de SMORN para este problema de agua subterránea.

Debido a que el acoplamiento entre MODFLOW y SMORN se hizo mediante archivos de texto plano usando awk y shell, este mismo código puede servir para resolver el mismo tipo de problema de control para cualquier modelo de MODFLOW, con solo substituir el nombre de los archivos que requieren ser leídos y editados: `Ultimo2.lst` y `Ultimo2.wel`, por los correspondientes en el nuevo modelo. En caso de querer resolver otro problema de control, sólo se necesitan editar las líneas donde se construyen las funciones objetivos, dentro del archivo `userDefinables.cpp`.

§4.5. Resultados

Lo primero que hay que mostrar es la forma en como método de optimización converge, lo cual se muestra en la Fig. §4.5 donde se grafican las funciones objetivo del bombeo total y el abatimiento. En la configuración original del bombeo, el total de extracción es de

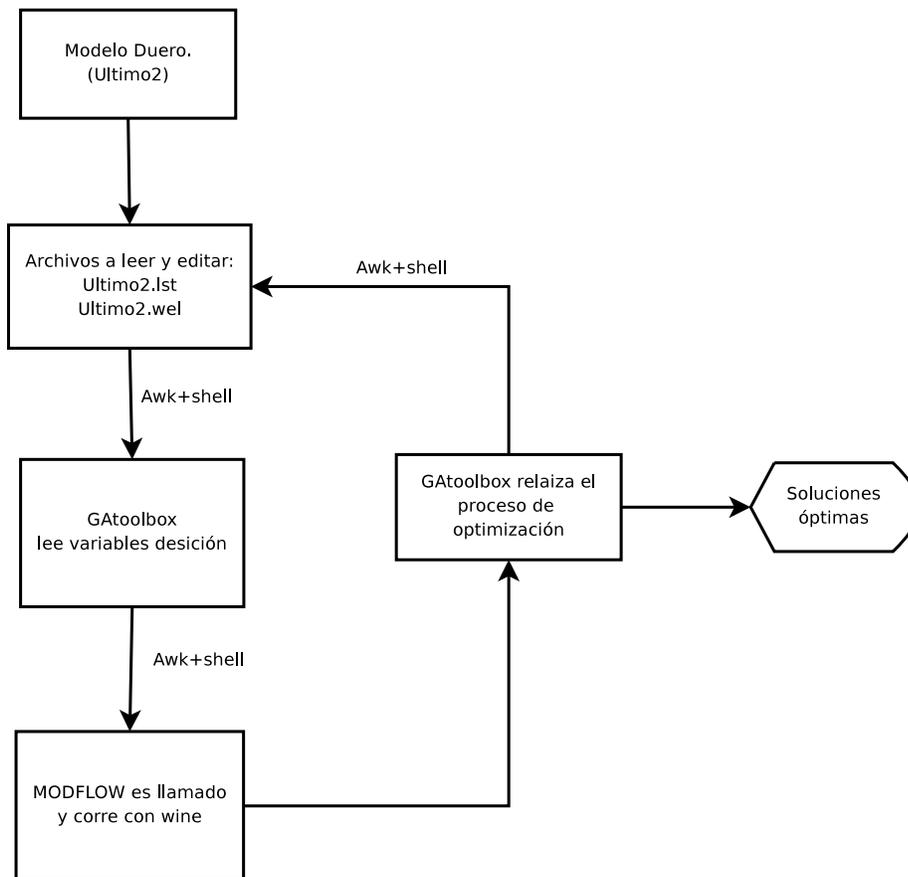


Figura 4.8

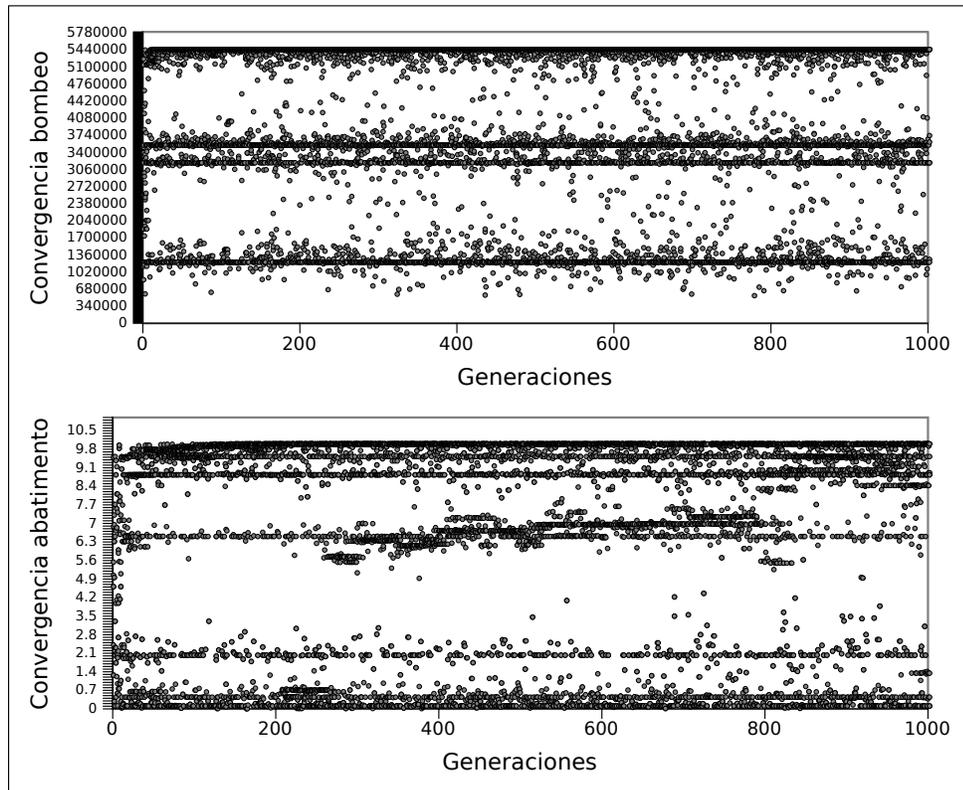


Figura 4.9: Convergencia de dos de las funciones objetivo, el bombeo total y el abatimiento promedio, usando los valores de la configuración original antes de la optimización.

$4.15 \times 10^5 m^3/s$, mientras que después de la optimización la solución con mayor valor de extracción total es de $5.45 \times 10^6 m^3/s$. Sin embargo, como analizaremos a detalle posteriormente, las soluciones conseguidas con el motor de optimización convergen a cuatro modas diferentes que mostramos en la siguiente figura.

Los valores mostrados a continuación son una normalización de los valores obtenidos directamente del proceso de optimización. En esta normalización el valor de uno corresponde al mejor resultado posible, mientras que el valor de cero es el peor. En el caso del bombeo total se buscó la solución óptima que diera el valor más grande y se dividieron todas las demás entre este valor. Para el abatimiento se realizó la misma normalización, pero la interpretación es opuesta a la del bombeo total debido a que la primera función objetivo es un máximo y la segunda un mínimo. De esta forma:

- Si el bombeo total es cercano a uno, se tiene una solución óptima donde se está extrayendo una cantidad muy grande del recurso, cercana a la mayor posible dado el problema.
- Si el bombeo total es cercano a cero, se tiene una solución óptima donde se está extra-

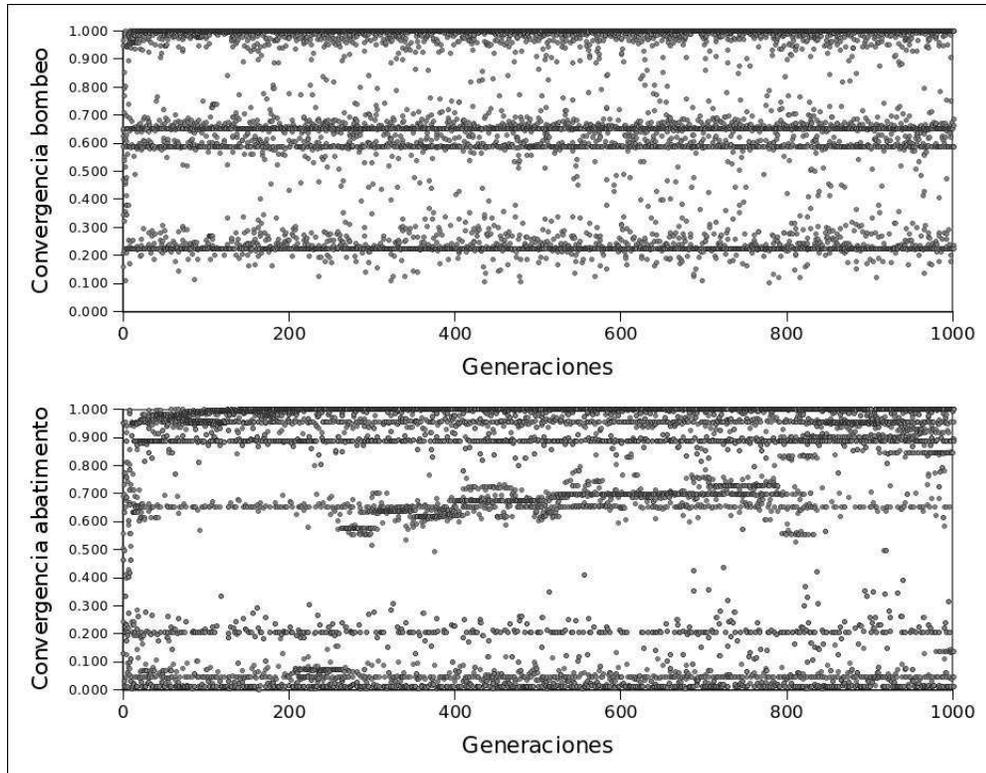


Figura 4.10: Convergencia de dos de las funciones objetivo, el bombeo total y el abatimiento promedio. Los valores están normalizados, de tal forma que uno corresponde al mejor valor obtenible, mientras que cero es el peor.

yendo muy poco recurso, cercana a la menor posible dado el problema.

- Si el abatimiento (promedio) es cercano a uno, se tiene una solución óptima donde hay muy poco abatimiento, cercano el menor posible dado el problema.
- Si el abatimiento es cercano a cero, se tiene una solución óptima donde hay un gran abatimiento, cercano al mayor posible dado el problema.

Estos valores normalizados también pueden considerarse como payoffs (ganancias) de un hipotético jugador que busca obtener el mayor beneficio de cualquiera de las funciones objetivo. De esta forma, mostrando el histograma de lo que ahora en adelante llamaremos los payoffs, en las figuras §4.5-§4.5, podemos identificar cuatro comportamientos bien diferenciados.

1. Soluciones óptimas en las cuales el compromiso entre las diferentes funciones objetivo privilegian la máxima extracción, sacrificando el abatimiento.
2. Soluciones óptimas en las cuales el compromiso entre las diferentes funciones objetivo privilegian la minimización del abatimiento, sacrificando el bombeo.

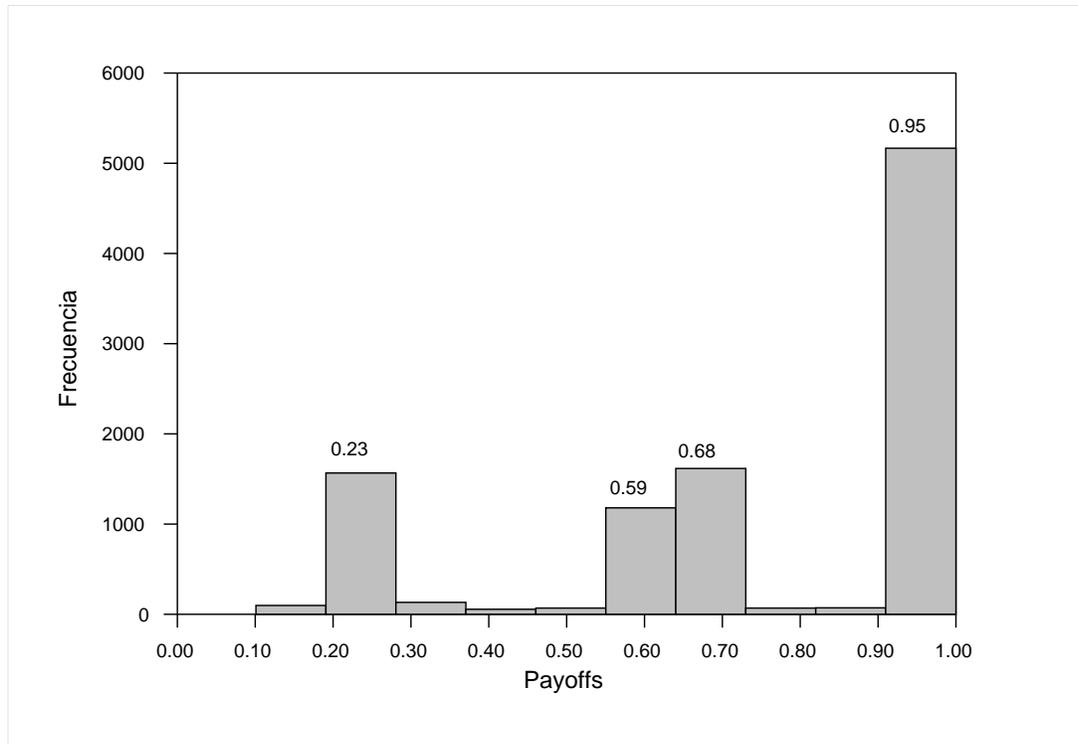


Figura 4.11: Se muestra el histograma de payoffs para el bombeo total. Se reconocen cuatro tipos de soluciones óptimas, que se marcan mostrando su valor de payoff.

3. Otros dos en los cuales ambas funciones objetivo están equilibradas, es decir no se extrae el máximo posible, pero tampoco se tiene el mayor abatimiento.

Otro resultado interesante se muestra en la Fig. §4.5 donde mostramos los histogramas de bombeo para los pozos de la cuenca. Como se puede apreciar, para los datos originales, es decir tal y como se está aprovechando el acuífero, la distribución de probabilidades se puede modelar por una normal, mientras que en el caso de la optimización es más bien una distribución uniforme. Este resultado es sumamente interesante desde el punto de vista estadístico, pues el hecho de la distribución de probabilidades pueda modelarse por una normal implica, que podemos encontrar pozos que sacan un volumen muy alto de agua, mientras otros sacan muy poco. Por el contrario, que después de la optimización la distribución de probabilidades sea uniforme, implica que el bombeo en el acuífero se está distribuyendo de forma equilibrada entre todos los pozos, es decir que no hay pozos que saquen demasiada agua mientras otros casi nada.

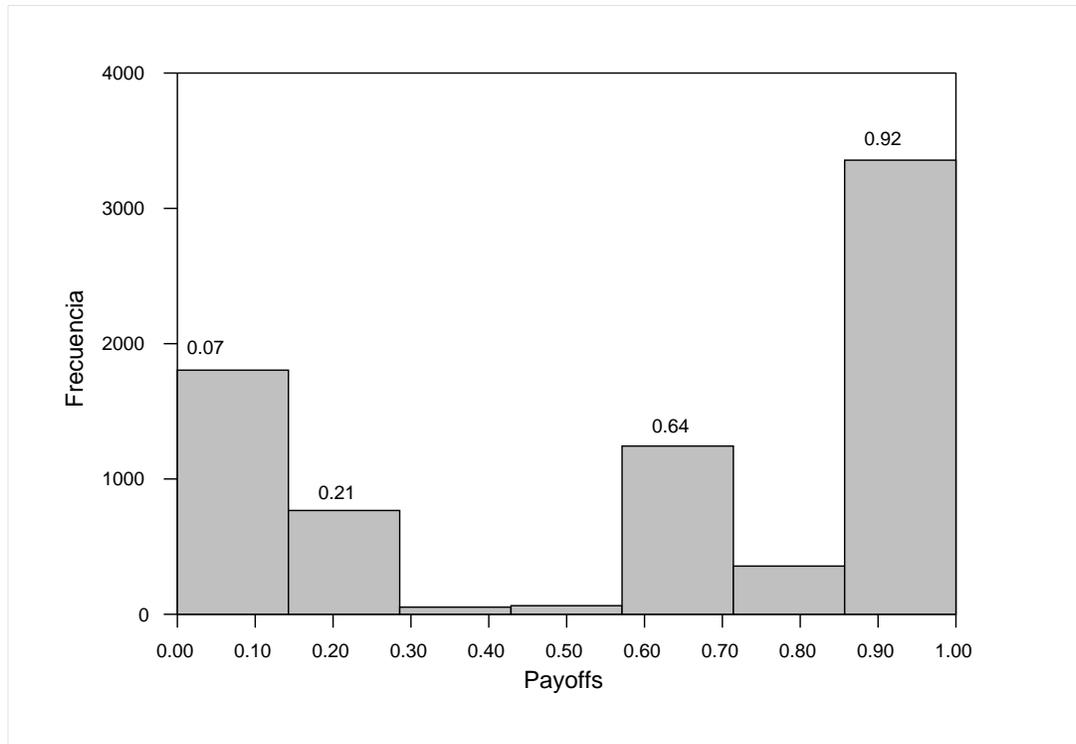


Figura 4.12: Se muestra el histograma de payoffs para el abatimiento. Se reconocen cuatro tipos de soluciones óptimas, que se marcan mostrando su valor de payoff.

§4.6. Discusión

En la sección anterior mostramos que en el estado actual el acuífero, en apariencia está siendo explotado de manera poco eficiente. Esto lleva a que haya una distribución socialmente desbalanceada de los recursos en la cual algunos usuarios extraen mucha agua, mientras otros solo obtienen el mínimo. Adicionalmente, uno puede inferir que esta situación también implica que la infraestructura de bombeo en algunos casos está sobre usada, mientras en otros se desperdicia. No obstante hay que advertir que una distribución uniforme de bombeo no siempre es asequible, ya sea por limitaciones de la infraestructura instalada, por los límites legales de los derechos de extracción o por restricciones hidrogeológicas.

Por otro lado, a pesar de que SMORN muestra una buena convergencia, emergen tres tipos de comportamientos todos ellos equivalentes desde el punto de vista de la optimización, lo cual hace que surja la pregunta ¿Qué tipo de solución debemos usar entonces?

Para contestar esta pregunta, recurrí al uso de la Teoría de Juegos, que es una formulación alterna de la Programación Dinámica, cuya teoría se presentó de forma resumida en el Capítulo 3. La formulación de la PD en forma de la Teoría de Juegos, permite estudiar interacciones en estructuras formalizadas de incentivos (los juegos y los payoffs), lo cual permite

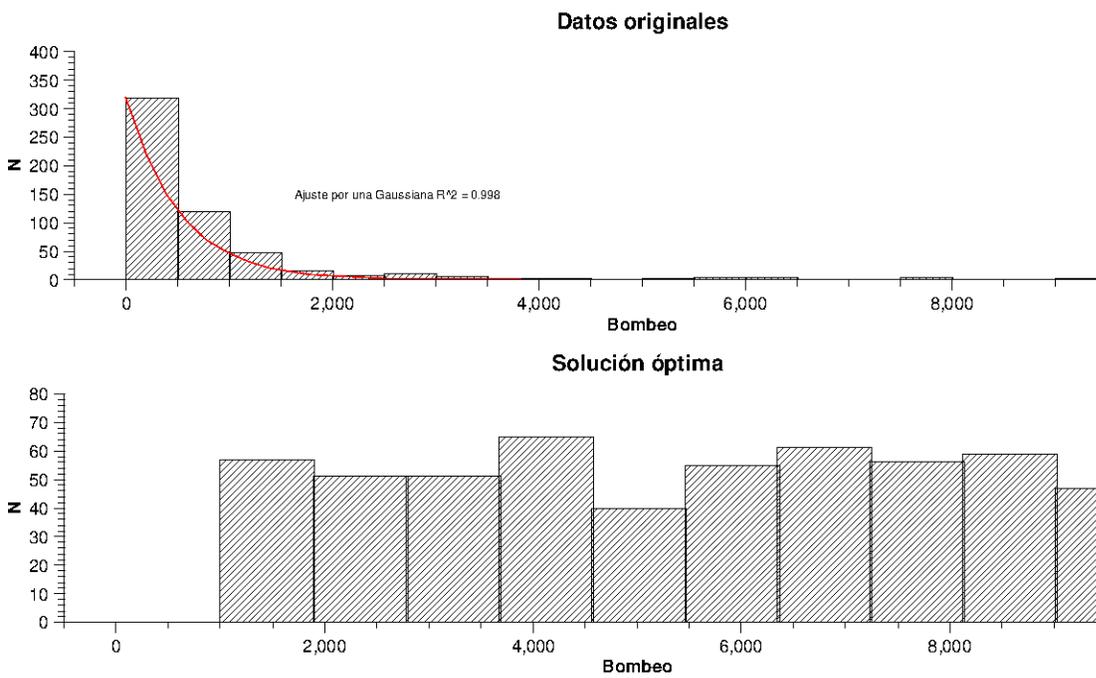


Figura 4.13: Se muestran los histogramas de los bombeos en la configuración original y en la configuración de una solución óptima, para mostrar la diferencia en la forma de uso del recurso.

entender el proceso de toma de decisiones. En general se estudian las estrategias óptimas así como el comportamiento previsto y observado de individuos en juegos. De esta forma, tipos de interacción aparentemente distintos pueden, en realidad, presentar estructuras de incentivos similares y, por lo tanto se pueden realizar generalizaciones que sirven para predecir lo que un jugador haría ante un juego particular.

Con esto en mente genere un juego extendido, que es una representación gráfica del juego en diagramas de árbol, donde cada nodo representa una situación donde el jugador debe tomar una decisión. Las líneas que parten de los nodos son acciones y en las terminaciones de las ramas se especifica el payoff correspondiente. En dicho juego hay cuatro jugadores:

- Un jugador llamado *usuarios*, que representa directamente a los usuarios, cuyo interés principal es obtener la mayor cantidad de recurso posible.
- Un jugador llamado *acuífero*, que representa a una parte de la sociedad que busca ante todo, la conservación y uso sostenible del acuífero.
- Un jugador llamado *admin*, que representa a un hipotético administrador del recurso natural, cuyo interés principal tener la mejor percepción pública posible.
- Un jugador llamado *chance (azar)*, que es invocado cuando una decisión debe ser tomada de forma completamente aleatoria.

De esta forma el Juego Acuífero, trata de plasmar la realidad de tomar decisiones cuando hay un grupo de intereses encontrados. Por un lado los usuarios desean que el administrador formule una política de manejo que les permita extraer la mayor cantidad de agua posible; por otro lado un sector de la sociedad, que estamos llamando acuífero, exigirá al administrador que sus políticas de manejo busquen conservar y hacer solo un uso sustentable del acuífero; finalmente el administrador quiere que ambos sectores de la sociedad lo perciban de la mejor manera posible, con fines electorales por ejemplo.

De esta forma construí un conjunto de payoffs que toman valores de cero a uno y representan:

- Para los usuarios, la proporción de agua que la política de manejo les permite sacar, respecto de un máximo posible y se obtiene del valor normalizado del bombeo total.
- Para el acuífero, representa el grado de conservación que la política de manejo esta considerando, a través de controlar el abatimiento y su velocidad y se obtiene promediando los valores normalizados de abatimiento promedio y velocidad de abatimiento promedio.

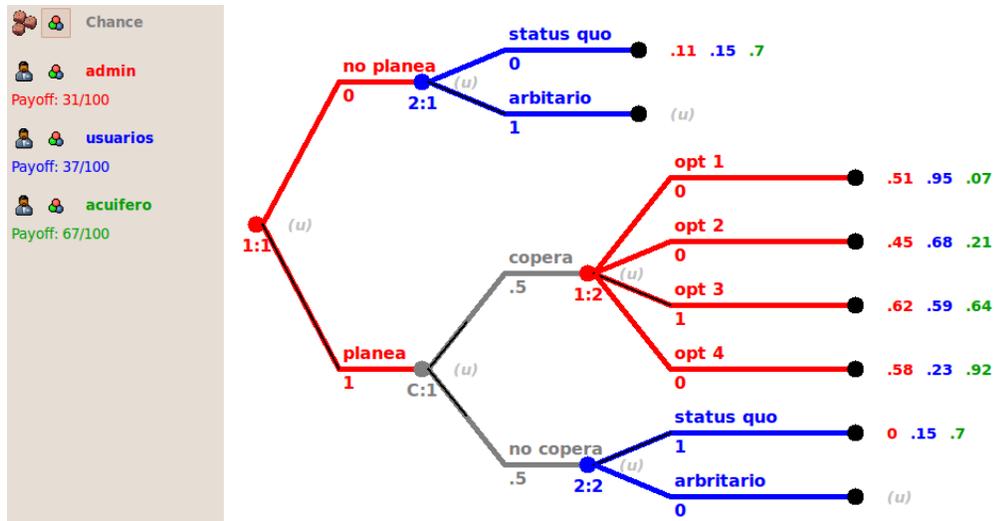


Figura 4.14: Juego extendido que representa el proceso de toma de decisión para elegir el tipo de solución óptima que más conviene, cuando hay intereses encontrados.

- Para el administrador, son el grado de conformidad (cuantas ganancias se obtienen respecto del máximo posible) que los otros jugadores tienen ante un plan de manejo particular y se calcula como el promedio de los payoffs de los usuarios y del acuífero.

Utilizando el software libre, con licencia GPL, Gambit [McKelvey, Richard D., McLennan, Andrew M., and Turocy, Theodore L. (2007) Gambit: Software Tools for Game Theory, Version 0.2007.01.30 <http://www.gambit-project.org>], construí el juego descrito arriba y que se muestra en la Fig. §4.6.

La primera jugada del juego esta a cargo de admin, quien puede decidir hacer o no, un plan de manejo para la cuenca. Si decide no hacer un plan de manejo, entonces los usuarios pueden decidir seguir utilizando el recurso como hasta ahora o bien hacer cambiar, pero como no hay un plan este cambio es en esencia aleatorio. En el primer caso se calcularon los payoffs usando el modelo con los datos originales, antes de la optimización. En el segundo caso no se asignan payoffs pues es una situación de la cual desconocemos su posible resultado.

Por otro lado, si admin decide hacer un plan de manejo, entonces puede ser que los usuarios decidan cooperar con el o no, situación que se representa por una movida del jugador azar. En la figura se muestra un par de valores de probabilidad particulares, pero se exploraron diferentes combinaciones. Otra interpretación de este par de valores, podría ser la proporción de usuarios que deciden cooperar con el administrados de la cuenca.

Ahora bien, en caso de que no cooperación, como sucedió en el primer nodo, las posibles opciones son seguir usando el recurso hasta ahora o cambiar a un uso, que por falta de plantación se considera arbitrario. Finalmente en el caso de cooperar, entonces las opciones son los cuatro tipo de soluciones a las que converge SMORN, una que privilegia a los usuarios,

uno que privilegia al acuífero y otras dos en las que hay un equilibrio entre extracción y conservación del acuífero.

Estructurado el juego de esta forma, se procedió a calcular los Equilibrios de Nash del juego estratégico asociado, para encontrar cuales serían las estrategias óptimas para cada jugador. En Teoría de Juegos, se define el equilibrio de Nash, como el conjunto de estrategias (óptimas) para cada jugador, que se alcanza cuando ninguno jugador puede beneficiarse al cambiar su estrategia mientras los otros no cambien la suya.

En este caso se encontraron dos Equilibrios de Nash, ambos equilibrios muestran que admin debería escoger hacer un plan de manejo óptimo, que los usuarios entonces en caso de no cooperar deben optar por no cambiar su forma de uso del recurso y que en caso de cooperar deben escoger usar el tipo de solución equilibrada, en particular la de menor bombeo total. Este resultado mostró ser robusto ante cualquier pareja de valores en la jugada del azar.

La diferencia entre ambos Equilibrios es que uno muestra que si admin decide no planear, entonces la mejor estrategia es mantener su forma de uso del recurso, mientras que el otro Equilibrio de Nash muestra que lo mejor sería cambiar.

Este análisis mediante Teoría de Juegos muestra su utilidad ante escenarios de conflicto, pero hay que recordar que solo nos responde que tipo de solución conviene adoptar, sin embargo cada tipo de solución tiene muchas soluciones alternativas. Esto más que verse como un problema es una bondad de haber resuelto el problema con computo evolutivo, pues dado que cada solución es equivalente a la otra, desde el punto de vista del control, entonces tenemos la opción de escoger de entre ellas la que queramos. Esto representa una magnífica oportunidad de negociación, pues quizás algunas de las soluciones generadas pueda afectar a algún usuario particular, quien querrá negociar con el administrador y los demás usuarios para no verse perjudicado.

§4.7. Conclusiones

Conforme la presión por el uso de recursos naturales se siga intensificando, los tomadores de decisiones tendrán que idear planes de manejo que no sólo sean capaces de tomar en cuenta varios objetivos, sino que además puedan resolver diferentes escenarios de conflicto entre usuarios que compiten por un mismo recurso.

En el caso del manejo del agua subterránea en la cuenca del Río Duero, se mostró que actualmente se está llevando a cabo un manejo prácticamente aleatorio del recurso. Debido a que la zona es económicamente importante, se prevé además que la demanda sobre el recurso se incremente considerablemente en el futuro, con lo cual surge la necesidad de generar un plan de manejo adecuado. De hecho, después del análisis usando SMORN, se mostró que bajo cualquier perspectiva, la mejor estrategia de las autoridades es efectivamente crear el plan de manejo que se requiere.

Aunque sencilla, la aplicación al problema de gestión de la cuenca del Río Duero, muestra que SMORN tiene un gran potencial para resolver problemas de control multiojetivo de recursos naturales en situaciones de conflicto.

Se determinó que el tipo de plan de manejo que se debe adoptar es uno en el cual tanto la extracción del recurso como el cuidado del acuífero están equilibrados. Además se hizo notar que debido a que las soluciones generadas por SMORN son equivalentes, hay un buen espacio a la negociación y resolución de conflictos. De esta forma SMORN muestra ser tanto una herramienta de optimización como de toma de decisiones y negociación.

Sin embargo queda claro que existe un número de mejoras que se podrían hacer a la aplicación para afinar los resultados obtenidos, como son:

- Establecer los intervalos de valores que toman las variables de decisión usando para calcular el límite superior: el volumen concesionado por CNA al pozo en cuestión, su capacidad instalada de bombeo y las restricciones hidrogeológicas; mientras que para el límite inferior habría que tomar lo que el usuario considera el volumen mínimo que requiere.
- Para que ese sistema pudiera ser ocupado en el mundo real, habría que implementar una GUI (Interfase gráfica de usuario) que permitiera a un usuario, grupo de usuarios o administrador, utilizar sus propios archivos de MODFLOW y correr el sistema, sin necesidad de editar archivo alguno. De igual forma sería muy deseable que el sistema estuviera disponible en una aplicación en línea.
- Para hacer accesible el sistema incluso a usuarios que no tengan elaborado un modelo en MODFLOW, se debería optar por usar el módulo GMT de Grass. Grass es un SIG

libre que al integrarle el modulo GMT, permite usar a MODFLOW embebido en su sistema. Una ventaja de esto es que Grass-GMT permite crear el modelo discreto del sistema a partir de modelos de elevación de forma gráfica y muy intuitiva.

- En la parte del análisis de las soluciones generadas usando Teoría de Juegos, sería interesante distinguir entre diferentes tipos de usuarios del recurso: agropecuario, industrial y urbano. Cada usuario con intereses particulares que pueden estar conflictuados con los de los demás.

§4.8. Perspectivas futuras.

Debido a los buenos resultados obtenidos en este estudio, ya estoy trabajando en una aplicación para el manejo óptimo del suelo. En particular estoy trabajando para identificar cual sería la locación óptima desde diferentes puntos de vista: edafológico, económico y conservación; para instalar un nuevo basurero municipal en Tuxpan, Michoacán. Para ello estoy usando una aproximación geomorfopedológica del uso del suelo, en la que se identifican las características edáficas del suelo por unidad geomorfológica, como se muestra en la Fig. §4.8.

De esta forma, a cada geoforma se le puede asignar una serie de características que determinan su “vocación” para ser usados. Estas características determinan la cantidad de agua contenida en el suelo, la cantidad de materia orgánica presente, disponibilidad de nutrientes, estabilidad geomorfológica, etc. Así, se puede definir una función objetivo como la distancia del vector características de una locación particular, al vector de características óptimas para el uso X, en este caso X: basurero. Una segunda función objetivo se puede construir con el costo de instalar y operación, la cual estará fuertemente influenciada por la distancia al pueblo, pues más que otros factores, son los costos de trasportación los que pueden de una muy directa diferenciar la viabilidad económica entre dos sitios. Finalmente, no podemos dejar de lado los impactos al medio ambiente que se generan por la presencia de un relleno sanitario; impactos que cambian dependiendo de las características y potenciales del suelo en donde se pretende construir. Por ejemplo, el posible impacto ambiental que se generaría sobre un suelo cuyo mayor potencial es de pastizal, es muy diferente del posible impacto en un suelo cuyo mayor potencial es la recarga del acuífero. En la Fig. §4.8 se muestran los potenciales para cada unidad geomorfológica en función de las características del suelo [reporte campo].

Los primeros resultados de SMORN, mostrados en la Fig. §4.8, determinan que el mejor sitio para instalar el nuevo basurero municipal desde la perspectiva geomorfopedológica, es el 73 en la nomenclatura de la implementación, que corresponde a una Geoforma de avalancha y subforma de zona baja cóncavo-coluvial. Lo cual esta en buen acuerdo con lo que se determino en estudios anteriores [reporte campo], ver Fig. §4.8.

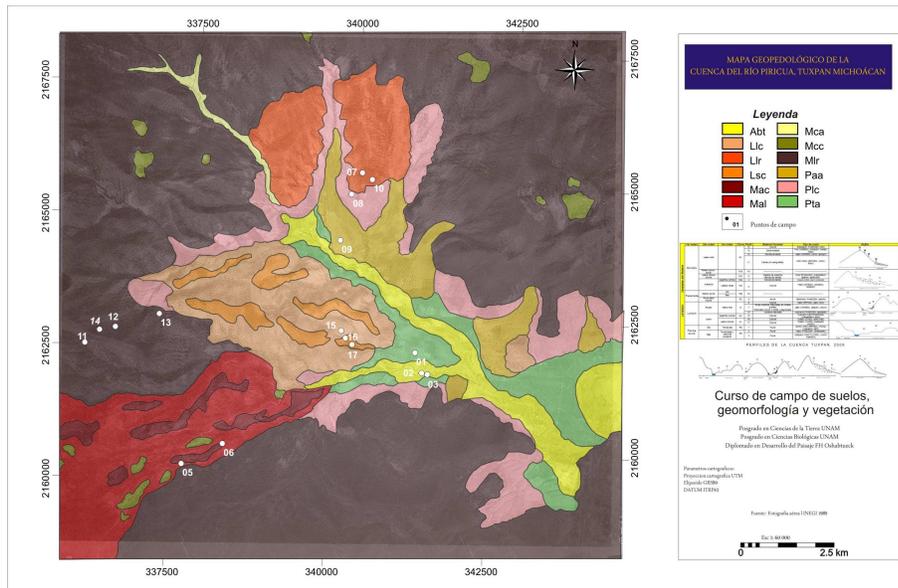


Figura 4.15: Mapa geomorfopedológico de Tuxpan, Michoacán. [reporte campo]

Potencial / Unidad	Planicie aluvial			Abanico	Lomerío suave			Lomerío abrupto			Avalancha			Montaña			
	T baj	T medi	T alta	Abanico aluvial	Pie ladera	Ladera	Cresta	Pie ladera	Lad. Norte	Lad. Sur	Sup. cumbra	Lad. mediana	Z. baja cóncava	Pie ladera	Lad. baja	Lad. alta	Cresta
	03	02	01	09	17	16	15	08	07	10	04	05	06	13	12	14	11
Aguacate	mA	mA	A	mA	mA	NA	NA	mA	NA	NA	A	NA	NA	A	mA	NA	NA
Guayaba	mA	A	A	mA	A	NA	NA	A	NA	NA	NA	A	mA	NA	NA	NA	NA
Maiz	A	MA	MA	mA	mA	NA	NA	mA	MA	NA	MA	NA	NA	mA	NA	NA	NA
Basurero	NA	NA	NA	mA	NA	mA	NA	mA	NA	NA	mA	NA	A	NA	NA	NA	NA
Recarga de acuífero	NA	NA	NA	NA	NA	NA	NA	mA	A	A	mA	A	mA	NA	MA	MA	MA
Conservación biológica	NA	NA	NA	NA	NA	NA	NA	NA	mA	mA	A	A	NA	NA	A	MA	MA
Manejo Forestal	NA	NA	NA	NA	NA	NA	NA	NA	A	A	MA	MA	NA	NA	MA	A	A
Plantaciones comerciales	NA	NA	NA	NA	A	MA	MA	A	mA	mA	MA	NA	NA	MA	NA	NA	NA
Captura de Carbono	NA	NA	mA	mA	mA	mA	mA	mA	A	mA	A	A	A	mA	MA	A	A
Pastizal / Ganadería	A	A	A	A/ mA	A/ mA	mA	A/ mA	A/ mA	NA	NA	A	NA	A	A	NA	NA	NA

Figura 4.16: Usos potenciales por unidad geomorfológica MA= muy apto, A=apto, mA= moderadamente apto, NA= no apto. Abreviaciones: T= terraza, Lad =ladera, Sup= superficie.

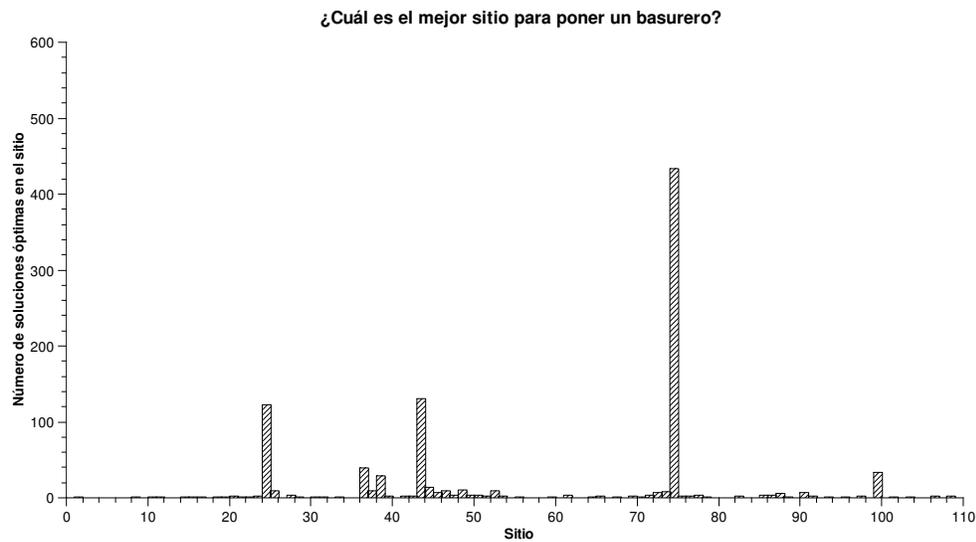


Figura 4.17: Histograma de soluciones.

No obstante estos primeros resultados son alentadores, hace falta todavía muchas mejoras en la implementación, razón por la cual no se incluyó formalmente en este trabajo.

Apéndices

Línea de tiempo.

1868

- J. C. Maxwell analiza la estabilidad del sistema de control en la máquina de Watt[55].

1877

- E. J. Routh desarrolla una técnica numérica para determinar cuando una ecuación característica tiene raíces estables [66].
- I. I. Vishnegradsky analiza, de forma independiente a Maxwell, la estabilidad de reguladores usando ecuaciones diferenciales [74].

1892

- A. M. Lyapunov estudia la estabilidad de ecuaciones diferenciales no lineales [53, 52].

1892-1898

- O. Heaviside desarrolla el calculo operacional, estudia el comportamiento de los transitorios y establece las bases del análisis con funciones de transferencia [52].

1893

- A. B. Stodola estudia la regulación de agua en turbinas usando las técnicas de Vishnegradsky [52].

1920-1939

- P. S. de Laplace (1749—1827), J. Fourier (1768—1830), A.L. Cauchy (1789—1857) desarrollan el análisis en el espacio de las frecuencias y exploran su aplicación en telecomunicaciones trabajando en proyectos de los laboratorios Bell [52].

1927

- H. S. Black demuestra la utilidad de las retroalimentaciones negativas [13].

1932

- H. Nyquist desarrolla la teoría de regeneración para el diseño de amplificadores estables, con lo cual genera el criterio de Nyquist para estabilidad [59].

1941

- A. C. Hall reconoce los efectos adversos de ignorar el ruido en el diseño de sistemas de control[40].

1942

- N. Wiener analiza los sistemas de procesamiento de información usando modelos estocásticos, desarrollando los filtros estadísticamente óptimos para señales estacionarias en tiempo continuo[75].

1958

- L. S. Pontryagin desarrolla su principio extremal, fundamentando la teoría de control óptimo [63].

A partir de aquí se considera el inicio de la teoría de control moderna.

GNU Free Documentation Licence

GNU Free Documentation License Version 1.3, 3 November 2008

Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, L^AT_EX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions

of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission. B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement. C. State on the Title page the name of the publisher of the Modified Version, as the publisher. D. Preserve all the copyright notices of the Document. E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices. F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below. G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice. H. Include an unaltered copy of this License. I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence. J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission. K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein. L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles. M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version. N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section. O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as

Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents

with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly

and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that

were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Códigos fuente SMORN.

```

/*****
/*Sistema de Manejo Optimo de Recursos Naturales (SMORN) */
/* wrote by Oliver López <lopezoliverx@otrasenda.org> Msc at IIMAS, UNAM.*/
/*Based on Single & Multi-Objective Genetic Algorithms Toolbox Code in C++ */
/* wrote by Kumara Sastry */
/* Illinois Genetic Algorithms Laboratory (IlligAL) */
/* Department of General Engineering */
/* University of Illinois at Urbana-Champaign */

##### input file #####
#Input file for the GA Toolbox
# Author: Oliver López
#
#
# GA type: SGA or NSGA
#
NSGA

```



```

#
NoNiching
#
# Selection
# Usage: Selection type, [parameter(s)...]
# To use the default setting type "default"
#
# Valid selection types and optional parameters are:
# RouletteWheel
# SUS
# TournamentWOR [tournament size]
# TournamentWR [tournament size]
# Truncation [# copies]
#
# When using NSGA, it can be neither SUS nor RouletteWheel.
#
TournamentWOR 2
#
# Crossover
# Crossover probability
# To use the default setting type "default"
#
# Usage: Crossover type, [parameter(s)...]
# To use the default crossover method type "default"
# Valid crossover types and optional parameters are
# OnePoint
# TwoPoint
# Uniform [gene-wise swap probability]
# SBX [gene-wise swap probability][order of the polynomial]
#
0.9
SBX 0.5 10
#
# Mutation
# Mutation probability
# To use the default setting type "default"
#
# Usage: Mutation type, [parameter(s)...]
# Valid mutation types and the optional parameters are:
# Selective
# Polynomial [order of the polynomial]
# Gene-wise [sigma for gene #1][sigma for gene #2]...[sigma for gene #ell]
#
0.1
Polynomial 20
#
# Scaling method
# To use the default setting type "default"
#
# Usage: Scaling method, [parameter(s)...]
# Valid scaling methods and optional parameters are:
# NoScaling
# Ranking
# SigmaScaling [scaling parameter]
#
NoScaling
#
# Constraint-handling method
# To use the default setting type "default"
#
# Usage: Constraint handling method, [parameters(s)...]
# Valid constraint handling methods and optional parameters are
# NoConstraints
# Tournament
# Penalty [Linear|Quadratic]
#
NoConstraints
#
# Local search method
# To use the default setting type "default"
#
# Usage: localSearchMethod, [maxLocalTolerance], [maxLocalEvaluations],
# [initialLocalPenaltyParameter], [localUpdateParameter],
# [lamarkianProbability], [localSearchProbability]

```

```

#
# Valid local search methods are: NoLocalSearch and SimplexSearch
#
# For example, SimplexSearch 0.001000 20 0.500000 2.000000 0.000000 0.000000
NoLocalSearch
#
# Stopping criteria
# To use the default setting type "default"
#
# Number of stopping criterias
#
# If the number is greater than zero
# Number of generation window
# Stopping criterion, Criterion parameter
#
# Valid stopping criterias and the associated parameters are
# NoOfEvaluations, Maximum number of function evaluations
# FitnessVariance, Minimum fitness variance
# AverageFitness, Maximum value
# AverageObjective, Max/Min value
# ChangeInBestFitness, Minimum change
# ChangeInAvgFitness, Minimum change
# ChangeInFitnessVar, Minimum change
# ChangeInBestObjective, Minimum change
# ChangeInAvgObjective, Minimum change
# NoOfFronts (NSGA only), Minimum number
# NoOfGuysInFirstFront (NSGA only), Minimum number
# ChangeInNoOfFronts (NSGA only), Minimum change
# BestFitness (SGA with NoNiching only), Maximum value
#
default
#
# Load the initial population from a file or not
# To use the default setting type "default"
#
# Usage: Load population (0|1)
#
# For example, if you want random initialization type 0
# On the other and if you want to load the initial population from a
# file, type
# 1 <population file name> [0|1]
#
# Valid options for "Load population" are 0/1
# If you type "1" you must specify the name of the file to load the
# population from. The second optional parameter which indicates
# whether to evaluate the individuals of the loaded population or not.
0
# Save the evaluated individuals to a file
#
# To use default setting type "default".
#
# Here by default all evaluated individuals are stored and you will be
# asked for a file name later when you run the executable.
#
# Usage: Save population (0|1)
# For example, if you don't want to save the evaluated solutions type 0
# On the other and if you want to save the evaluated solutions
# 1 <save file name>
#
# Note that the evaluated solutions will be appended to the file.
#
# Valid options for "Save population" are 0/1
# If you type "1" you must specify the name of the file to save the
# population to.
1 evaluatedSolutions.txt

/*****mean_draw.cpp*****/
#include <iostream>
#include <iomanip>
#include <math.h>
#include <stdio.h>
#include <cmath>
#include <fstream>
#include <stdlib.h>

```

```

#include <string.h>
#include <cstring>
#include <sstream>
using namespace std;
int main ()
{
int i,j;
draw1[]
return 0;
}

/*****read_write_wel_data.cpp*****/
#include <stdio.h>
#include <iostream>
#include <cmath>
#include <fstream>
#include <stdlib.h>
#include <iomanip>
using namespace std;
int i,j;
float Matrix[545][4];
float bombeo[545];
FILE *fpt, *fpt2;
main()
{
//***** reading data from ULTIMO_2.wel *****/
system("./calling_wel_shell.x");
system("./calling_wel_shell_2.x");
for (i = 0; i < 545; i++)
{
for (j = 0; j < 3; j++)
{
Matrix[i][j] = 0.0;
}
bombeo[i]=0.0;
}
fpt = fopen("data.wel","r");
if (fpt == NULL)
{
printf("\nError: File couldn't be opened (1).\n");
return -1;
}
fpt2 = fopen("bombeo.dat","r");
if (fpt2 == NULL)
{
printf("\nError: File couldn't be opened (1).\n");
return -1;
}
for(j = 0; j < 545; j++)
{
for(i = 0; i < 3; i++)
{
fscanf (fpt, "%f",&Matrix[j][i]);
}
fscanf (fpt2, "%f",&bombeo[j]);
}
fclose(fpt);
fclose(fpt2);
//***** example of operation over pumping data *****/
for(j = 0; j < 545; j++)
{
bombeo[j]=bombeo[j]/1000;
}
//***** opening and writing new wel file *****/
ofstream myfile ("ULTIMO_2.wel");
myfile << " 451 0 MXACTW IWELCB<< "\n" << " 93 0 ITMP NP -- Stress Period 1" << "\n" << "SFAC 5.348000E-01" << "\n";
myfile.setf (ios::left);
myfile.fill();
myfile.precision(3);
//***** writing stress period 1 *****/
for (i=0; i<93; i++)
{
if (Matrix[i][1]>10)
{

```

```

if (Matrix[i][2]<10)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else if (Matrix[i][2]>99)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
myfile.setf(ios::scientific, ios::floatfield);
myfile << " " << setiosflags(ios::uppercase) << scientific << bombeo[i];
myfile.unsetf(ios::scientific);
myfile << "\n";
}
else
{
if (Matrix[i][2]<10)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else if (Matrix[i][2]>99)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
myfile.setf(ios::scientific, ios::floatfield);
myfile << " " << setiosflags(ios::uppercase) << scientific << bombeo[i];
myfile.unsetf(ios::scientific);
myfile << "\n";
}
}
//***** writing stress period 2 *****//
myfile << " 451 0 ITMP NP -- Stress Period 2" << "\n" << "SFAC 1.306700E+00" << "\n";
for (i=93; i<544; i++)
{
if (Matrix[i][1]>9)
{
if (Matrix[i][2]<10)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else if (Matrix[i][2]>99)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
myfile.setf(ios::scientific, ios::floatfield);
myfile << " " << setiosflags(ios::uppercase) << scientific << bombeo[i];
myfile.unsetf(ios::scientific);
myfile << "\n";
}
else
{
if (Matrix[i][2]<10)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else if (Matrix[i][2]>99)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
}
}
}

```

```

myfile.setf(ios::scientific, ios::floatfield);
myfile << " " << setiosflags(ios::uppercase) << scientific << bombeo[i];
myfile.unsetf(ios::scientific);
myfile << "\n";
}
}
myfile.close();
return 0;
}
//***** reading data from ULTIMO_2.wel *****/
system("./calling_wel_shell.x");
system("./calling_wel_shell_2.x");
for (i = 0; i < 545; i++)
{
for (j = 0; j < 3; j++)
{
Matrix[i][j] = 0.0;
}
bombeo[i]=0.0;
}
fpt = fopen("data.wel","r");
if (fpt == NULL)
{
printf("\nError: File couldn't be opened (1).\n");
return -1;
}
fpt2 = fopen("bombeo.dat","r");
if (fpt2 == NULL)
{
printf("\nError: File couldn't be opened (1).\n");
return -1;
}
for(j = 0; j < 545; j++)
{
for(i = 0; i < 3; i++)
{
fscanf(fpt, "%f",&Matrix[j][i]);
}
fscanf(fpt2, "%f",&bombeo[j]);
}
fclose(fpt);
fclose(fpt2);
//***** example of operation over pumping data *****/
for(j = 0; j < 545; j++)
{
bombeo[j]=bombeo[j]/1000;
}
//***** opening and writing new wel file *****/
ofstream myfile ("ULTIMO_2.wel");
myfile << " 451 0 MIACTW IWELCB" << "\n" << " 93 0 ITMP NP -- Stress Period 1" << "\n" << "SFAC 5.348000E-01" << "\n";
myfile.setf(ios::left);
myfile.fill();
myfile.precision(3);
//***** writing stress period 1 *****/
for (i=0; i<93; i++)
{
if (Matrix[i][1]>10)
{
if (Matrix[i][2]<10)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else if (Matrix[i][2]>99)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
}
myfile.setf(ios::scientific, ios::floatfield);
myfile << " " << setiosflags(ios::uppercase) << scientific << bombeo[i];
myfile.unsetf(ios::scientific);
myfile << "\n";
}
}

```

```

}
else
{
if (Matrix[i][2]<10)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else if (Matrix[i][2]>99)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
myfile.setf(ios::scientific, ios::floatfield);
myfile << " " << setiosflags(ios::uppercase) << scientific << bombeo[i];
myfile.unsetf(ios::scientific);
myfile << "\n";
}
}
//***** writing stress period 2 *****/
myfile << " 451 0 ITMP NP -- Stress Period 2" << "\n" << "SFAC 1.306700E+00" << "\n";
for (i=93; i<544; i++)
{
if (Matrix[i][1]>9)
{
if (Matrix[i][2]<10)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else if (Matrix[i][2]>99)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
myfile.setf(ios::scientific, ios::floatfield);
myfile << " " << setiosflags(ios::uppercase) << scientific << bombeo[i];
myfile.unsetf(ios::scientific);
myfile << "\n";
}
else
{
if (Matrix[i][2]<10)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else if (Matrix[i][2]>99)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
myfile.setf(ios::scientific, ios::floatfield);
myfile << " " << setiosflags(ios::uppercase) << scientific << bombeo[i];
myfile.unsetf(ios::scientific);
myfile << "\n";
}
}
myfile.close();
return 0;
}

/*****userDefinables_mult.cpp*****/
#include <iostream>
#include <iomanip>
#include "ga.hpp"
#include "random.hpp"
#include "globalSetup.hpp"

```

```

#include <math.h>
#include <stdio.h>
#include <cmath>
#include <fstream>
#include <stdlib.h>
#include <string.h>
#include <cstring>
#include <sstream>
#define BLANK_STR " \t\n\r"
using namespace std;
GlobalSetup *globalSetup;
Random myRandom;
/* Objective function and constraints go here */
void globalEvaluate(double *x, double *objArray, double *constraintViolation,
double *penalty, int *noOfViolations)
{
int ii;
FILE *outEvals;
int i,j;
double Matrix[545][3];
double bombeo[545];
double balance[1];
double mean[9], draw[9], vel_draw[8];
double sum, vel_mean;
FILE *fpt, *fpt2, *fpt3, *fpt01, *fpt02, *fpt03, *fpt04, *fpt05, *fpt06;
FILE *fpt07, *fpt08, *fpt09;
for(ii = 0; ii < globalSetup->finalNoOfObjectives; ii++)
objArray[ii] = 0.0;
//***** single objective plus constraints *****//
//***** reading data from ULTIMO_2.wel *****//
if(globalSetup->gaType == SGA)
{
system("./mf2k.exe ULTIMO_2"); //calling modflow to generate ULTIMO_2.wel
//***** reading data from ULTIMO_2.wel *****//
system("sh read_wel.sh");
for (i = 0; i < 545; i++)
{
for (j = 0; j < 3; j++)
{
Matrix[i][j] = 0.0;
}
}
for(i=0; i<545; i++)
{
bombeo[i]=0.0;
}
for(i=0; i<1; i++)
{
balance[i]=0.0;
}
fpt = fopen("data.wel","r");
if (fpt == NULL)
{
printf("\nError: File couldn't be opened (1).\n");
}
fpt2 = fopen("bombeo.dat","r");
if (fpt2 == NULL)
{
printf("\nError: File couldn't be opened (1).\n");
}
for(i = 0; i < 545; i++)
{
for(j = 0; j < 3; j++)
{
fscanf(fpt, "%lf",&Matrix[i][j]);
}
}
for(i=0; i<545; i++)
{
fscanf(fpt2, "%lf",&bombeo[i]);
}
fclose(fpt);
fclose(fpt2);
//***** optimization part *****//

```

```

for(j = 0; j < 545; j++)
{
bombeo[j]=x[0];
}
sum = 0.0;
for (i = 0; i < 545; i++)
{
sum = sum + bombeo[i];
}
sum=sum*-1;
*objArray = sum; //this is the objective function !!!!
balance[0]=0;
system("sh read_balance_constrain.sh");
fpt3 = fopen("balance_constrain.dat","r");
fscanf(fpt3, "%lf", &balance[0]);
fclose(fpt3);
constraintViolation[0] = balance[0] - 712704;
if (constraintViolation[0] >= 0.0) constraintViolation[0] = 0.0;
system("sh read_draw_lay2.sh");
for(i=0;i<9;i++)
{
mean[i] = 0.0;
}
fpt01 = fopen("draw_lay2_step1_per2.dat","r");
fscanf(fpt01, "%lf", &mean[0]);
fclose(fpt01);
fpt02 = fopen("draw_lay2_step2_per2.dat","r");
fscanf(fpt02, "%lf", &mean[1]);
fclose(fpt02);
fpt03 = fopen("draw_lay2_step3_per2.dat","r");
fscanf(fpt03, "%lf", &mean[2]);
fclose(fpt03);
fpt04 = fopen("draw_lay2_step4_per2.dat","r");
fscanf(fpt04, "%lf", &mean[3]);
fclose(fpt04);
fpt05 = fopen("draw_lay2_step5_per2.dat","r");
fscanf(fpt05, "%lf", &mean[4]);
fclose(fpt05);
fpt06 = fopen("draw_lay2_step6_per2.dat","r");
fscanf(fpt06, "%lf", &mean[5]);
fclose(fpt06);
fpt07 = fopen("draw_lay2_step7_per2.dat","r");
fscanf(fpt07, "%lf", &mean[6]);
fclose(fpt07);
fpt08 = fopen("draw_lay2_step8_per2.dat","r");
fscanf(fpt08, "%lf", &mean[7]);
fclose(fpt08);
fpt09 = fopen("draw_lay2_step9_per2.dat","r");
fscanf(fpt09, "%lf", &mean[8]);
fclose(fpt09);
for(i=0;i<9;i++)
{
draw[i] = 0.0;
}
for(i=0; i<10; i++)
{
draw[0]=draw[0] + mean[i];
}
draw[0]=draw[0]*(-1);
constraintViolation[1] = draw[0] - 1.5;
if (constraintViolation[0] <= 0.0) constraintViolation[0] = 0.0;
for(i=0;i<7;i++)
{
vel_draw[i]=0.0;
}
for(i=0;i<7;i++)
{
vel_draw[i]=mean[i+1]-mean[i];
}
for(i=0;i<7;i++)
{
vel_draw[i]=vel_draw[i]*(-1);
}
constraintViolation[2] = vel_draw[0] - 1.0;

```

```

if (constraintViolation[0] <= 0.0) constraintViolation[0] = 0.0;
constraintViolation[3] = vel_draw[1] - 1.0;
if (constraintViolation[0] <= 0.0) constraintViolation[0] = 0.0;
constraintViolation[4] = vel_draw[2] - 1.0;
if (constraintViolation[0] <= 0.0) constraintViolation[0] = 0.0;
constraintViolation[5] = vel_draw[3] - 1.0;
if (constraintViolation[0] <= 0.0) constraintViolation[0] = 0.0;
constraintViolation[6] = vel_draw[4] - 1.0;
if (constraintViolation[0] <= 0.0) constraintViolation[0] = 0.0;
constraintViolation[7] = vel_draw[5] - 1.0;
if (constraintViolation[0] <= 0.0) constraintViolation[0] = 0.0;
constraintViolation[8] = vel_draw[6] - 1.0;
if (constraintViolation[0] >= 0.0) constraintViolation[0] = 0.0;
constraintViolation[9] = vel_draw[7] - 1.0;
if (constraintViolation[0] <= 0.0) constraintViolation[0] = 0.0;
//***** opening and writing new wel file *****//
ofstream myfile ("ULTIMO_2.wel");
myfile << " 451 0 MACTW IWELCB" << "\n" << " 93 0 ITMP NP -- Stress Period 1" << "\n" << "SFAC 5.348000E-01" << "\n";
myfile.setf (ios::left);
myfile.fill ();
myfile.precision (3);
//***** writing stress period 1 *****//
for (i=0; i<93; i++)
{
if (Matrix[i] [1]>10)
{
if (Matrix[i] [2]<10)
{
myfile << " " << Matrix[i] [0] << " " << Matrix[i] [1] << " " << Matrix[i] [2];
}
else if (Matrix[i] [2]>99)
{
myfile << " " << Matrix[i] [0] << " " << Matrix[i] [1] << " " << Matrix[i] [2];
}
else
{
myfile << " " << Matrix[i] [0] << " " << Matrix[i] [1] << " " << Matrix[i] [2];
}
myfile.setf (ios::scientific, ios::floatfield);
myfile << " " << setiosflags (ios::uppercase) << scientific << bombeo[i];
myfile.unsetf (ios::scientific);
myfile << "\n";
}
else
{
if (Matrix[i] [2]<10)
{
myfile << " " << Matrix[i] [0] << " " << Matrix[i] [1] << " " << Matrix[i] [2];
}
else if (Matrix[i] [2]>99)
{
myfile << " " << Matrix[i] [0] << " " << Matrix[i] [1] << " " << Matrix[i] [2];
}
else
{
myfile << " " << Matrix[i] [0] << " " << Matrix[i] [1] << " " << Matrix[i] [2];
}
myfile.setf (ios::scientific, ios::floatfield);
myfile << " " << setiosflags (ios::uppercase) << scientific << bombeo[i];
myfile.unsetf (ios::scientific);
myfile << "\n";
}
}
//***** writing stress period 2 *****//
myfile << " 451 0 ITMP NP -- Stress Period 2" << "\n" << "SFAC 1.306700E+00" << "\n";
for (i=93; i<544; i++)
{
if (Matrix[i] [1]>9)
{
if (Matrix[i] [2]<10)
{
myfile << " " << Matrix[i] [0] << " " << Matrix[i] [1] << " " << Matrix[i] [2];
}
else if (Matrix[i] [2]>99)

```

```

{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
myfile.setf(ios::scientific, ios::floatfield);
myfile << " " << setiosflags(ios::uppercase) << scientific << bombeo[i];
myfile.unsetf(ios::scientific);
myfile << "\n";
}
else
{
{
if (Matrix[i][2]<10)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else if (Matrix[i][2]>99)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
myfile.setf(ios::scientific, ios::floatfield);
myfile << " " << setiosflags(ios::uppercase) << scientific << bombeo[i];
myfile.unsetf(ios::scientific);
myfile << "\n";
}
}
}
myfile.close();
}
//***** more than one objective funtions non edited from original example *****//*****//*****//*****//*****//
else
{
system("./mf2k.exe ULTIMO_2"); //calling modflow to generate ULTIMO_2.wel
//***** reading data from ULTIMO_2.wel *****//*****//*****//*****//*****//
system("sh read_wel.sh");
for (i = 0; i < 545; i++)
{
for (j = 0; j < 3; j++)
{
Matrix[i][j] = 0.0;
}
}
for(i=0; i<545; i++)
{
bombeo[i]=0.0;
}
for(i=0; i<1; i++)
{
balance[i]=0.0;
}
fpt = fopen("data.wel","r");
if (fpt == NULL)
{
printf("\nError: File couldn't be opened (1).\n");
}
fpt2 = fopen("bombeo.dat","r");
if (fpt2 == NULL)
{
printf("\nError: File couldn't be opened (1).\n");
}
for(i = 0; i < 545; i++)
{
for(j = 0; j < 3; j++)
{
fscanf(fpt, "%lf",&Matrix[i][j]);
}
}
for(i=0; i<545; i++)
{

```

```

fscanf(fpt2, "%lf",&bombero[i]);
}
fclose(fpt);
fclose(fpt2);
//***** optimization part *****/
for(j = 0; j < 545; j++)
{
bombero[j]=x[0];
}
sum = 0.0;
for (i = 0; i < 545; i++)
{
sum = sum + bombero[i];
}
sum=sum*-1;
objArray[0] = sum; //this is the objective function !!!!
system("sh read_draw_lay2.sh");
for(i=0;i<9;i++)
{
mean[i] = 0.0;
}
fpt01 = fopen("draw_lay2_step1_per2.dat","r");
fscanf(fpt01, "%lf", &mean[0]);
fclose(fpt01);
fpt02 = fopen("draw_lay2_step2_per2.dat","r");
fscanf(fpt02, "%lf", &mean[1]);
fclose(fpt02);
fpt03 = fopen("draw_lay2_step3_per2.dat","r");
fscanf(fpt03, "%lf", &mean[2]);
fclose(fpt03);
fpt04 = fopen("draw_lay2_step4_per2.dat","r");
fscanf(fpt04, "%lf", &mean[3]);
fclose(fpt04);
fpt05 = fopen("draw_lay2_step5_per2.dat","r");
fscanf(fpt05, "%lf", &mean[4]);
fclose(fpt05);
fpt06 = fopen("draw_lay2_step6_per2.dat","r");
fscanf(fpt06, "%lf", &mean[5]);
fclose(fpt06);
fpt07 = fopen("draw_lay2_step7_per2.dat","r");
fscanf(fpt07, "%lf", &mean[6]);
fclose(fpt07);
fpt08 = fopen("draw_lay2_step8_per2.dat","r");
fscanf(fpt08, "%lf", &mean[7]);
fclose(fpt08);
fpt09 = fopen("draw_lay2_step9_per2.dat","r");
fscanf(fpt09, "%lf", &mean[8]);
fclose(fpt09);
for(i=0;i<9;i++)
{
draw[i] = 0.0;
}
for(i=0; i<10; i++)
{
draw[0]=draw[0] + mean[i];
}
draw[0]=draw[0]*(-1);
//second objective function//
draw[0]=x[545];
objArray[1] = draw[0];
//third objective function///
for(i=0;i<7;i++)
{
vel_draw[i]=0.0;
}
for(i=0;i<7;i++)
{
vel_draw[i]=mean[i+1]-mean[i];
}
for(i=0;i<7;i++)
{
vel_draw[i]=vel_draw[i]*(-1);
}
vel_mean=(vel_draw[0]+vel_draw[1]+vel_draw[2]+vel_draw[3]+vel_draw[4]+vel_draw[5]+vel_draw[6]+vel_draw[7])/8;

```

```

vel_mean=x[546];
objArray[2] = vel_mean;
//***** opening and writing new wel file *****//
ofstream myfile ("ULTIMO_2.wel");
myfile << " 451 0 MxACTW IWELCB" << "\n" << " 93 0 ITMP NP -- Stress Period 1" << "\n" << "SFAC 5.348000E-01" << "\n";
myfile.setf(ios::left);
myfile.fill();
myfile.precision(3);
//***** writing stress period 1 *****//
for (i=0; i<93; i++)
{
if (Matrix[i][1]>10)
{
if (Matrix[i][2]<10)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else if (Matrix[i][2]>99)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
myfile.setf(ios::scientific, ios::floatfield);
myfile << " " << setiosflags(ios::uppercase) << scientific << bombeo[i];
myfile.unsetf(ios::scientific);
myfile << "\n";
}
else
{
if (Matrix[i][2]<10)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else if (Matrix[i][2]>99)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
myfile.setf(ios::scientific, ios::floatfield);
myfile << " " << setiosflags(ios::uppercase) << scientific << bombeo[i];
myfile.unsetf(ios::scientific);
myfile << "\n";
}
}
//***** writing stress period 2 *****//
myfile << " 451 0 ITMP NP -- Stress Period 2" << "\n" << "SFAC 1.306700E+00" << "\n";
for (i=93; i<544; i++)
{
if (Matrix[i][1]>9)
{
if (Matrix[i][2]<10)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else if (Matrix[i][2]>99)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
myfile.setf(ios::scientific, ios::floatfield);
myfile << " " << setiosflags(ios::uppercase) << scientific << bombeo[i];
myfile.unsetf(ios::scientific);
myfile << "\n";
}
else

```

```

{
if (Matrix[i][2]<10)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else if (Matrix[i][2]>99)
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
else
{
myfile << " " << Matrix[i][0] << " " << Matrix[i][1] << " " << Matrix[i][2];
}
myfile.setf(ios::scientific, ios::floatfield);
myfile << " " << setiosflags(ios::uppercase) << scientific << bombo[i];
myfile.unsetf(ios::scientific);
myfile << "\n";
}
}
myfile.close();
}
// ***** end of objective editing, rest stay non edited *****//
*penalty = 0.0;
*noOfViolations = 0;
for(ii = 0; ii < globalSetup->finalNoOfConstraints; ii++) {
if (constraintViolation[ii] != 0) {
if (globalSetup->constraintMethod == Penalty) {
if (globalSetup->penaltyFunction == Linear)
*penalty += globalSetup->penaltyWeights[ii]*fabs(constraintViolation[ii]);
else if (globalSetup->penaltyFunction == Quadratic)
*penalty += globalSetup->penaltyWeights[ii]*
(constraintViolation[ii]*constraintViolation[ii]);
}
else
*penalty += fabs(constraintViolation[ii]);
}
if (globalSetup->savePopulation) {
outEvals = fopen(globalSetup->saveEvalSolutions, "a");
for(ii = 0; ii < globalSetup->noOfDecisionVariables; ii++) {
fprintf(outEvals, "%f\t", x[ii]);
}
for(ii = 0; ii < globalSetup->finalNoOfObjectives; ii++) {
fprintf(outEvals, "%f\t", objArray[ii]);
}
if (globalSetup->finalNoOfConstraints > 0) {
for(ii = 0; ii < globalSetup->finalNoOfConstraints; ii++) {
fprintf(outEvals, "%f\t", constraintViolation[ii]);
}
}
fprintf(outEvals, "%f", *penalty);
}
fprintf(outEvals, "\n");
fflush(outEvals);
fclose(outEvals);
}
}
/*
Read a non-comment, non-blank line from the specified file stream
At EOF, it returns NULL.
Otherwise, it returns the pointer to the first token.
(The string just read in is altered by strtok().)
*/
static char* readOneLine(char *pcBuf, int iMaxSize, FILE *fStream)
{
char *pToken;
do {
pToken = NULL;
*pcBuf = '\0';
fgets(pcBuf, iMaxSize, fStream);
if (feof(fStream))
break;
// get the first token
pToken = strtok(pcBuf, BLANK_STR);
// if there is no first token, it is a blank line.
// if the first token starts with '#', it is a comment line.

```

```

} while ((pToken == NULL) || (*pToken == '#'));
return (pToken);
}
int main(int argc, char *argv[]) {
int ii;
const int ciBufSize = 1024;
char *pToken, caBuf[ciBufSize];
FILE *fInput, *fOutput;
if (argc != 2) {
printf("Error! Usage is GAtbx inputfile\n");
exit(1);
}
fInput = fopen(argv[1], "r");
if (fInput == NULL) {
printf("Error! opening file%s\n", argv[1]);
exit(1);
}
globalSetup = new GlobalSetup;
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
//GA type
if (strcmp("SGA", pToken) == 0) {
globalSetup->gaType = SGA;
}
else if (strcmp("NSGA", pToken) == 0) {
globalSetup->gaType = NSGA;
}
else {
fclose(fInput);
printf("Unknown parameter! It should be either SGA or NSGA\n");
exit(1);
}
// decision variables
// read the number of decision variables
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
// the number can't be less than or equal to zero
if ((globalSetup->noOfDecisionVariables = atoi(pToken)) <= 0) {
fclose(fInput);
printf("Error! number of decision variables should be > 0\n");
exit(1);
}
globalSetup->variableTypes = new VariableType[globalSetup->noOfDecisionVariables];
globalSetup->variableRanges = new double*[(globalSetup->noOfDecisionVariables)];
for(ii = 0; ii < globalSetup->noOfDecisionVariables; ii++) {
// read a line
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
// variable type
if (strcmp("double", pToken) == 0) {
globalSetup->variableTypes[ii] = Real;
}
else if (strcmp("int", pToken) == 0) {
globalSetup->variableTypes[ii] = Integer;
}
else {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
// variable ranges
// allocate memory
globalSetup->variableRanges[ii] = new double[2];
// lower bound
if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {

```

```

fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
globalSetup->variableRanges[ii][0] = atof(pToken);
// upper bound
if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
globalSetup->variableRanges[ii][1] = atof(pToken);
if (globalSetup->variableRanges[ii][1] <= globalSetup->variableRanges[ii][0]) {
fclose(fInput);
printf("Error! lower bound, %f, must be lower than the upper bound, %f\n", globalSetup->variableRanges[ii][0], globalSetup->variableRanges[ii][1]);
exit(1);
}
}
// objectives
// read the number of objectives
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
// the number can't be less than or equal to zero
if ((globalSetup->noOfRawObjectives = atoi(pToken)) <= 0) {
fclose(fInput);
printf("Error! number of objectives should be > 0\n");
exit(1);
}
globalSetup->finalNoOfObjectives = globalSetup->noOfRawObjectives;
globalSetup->noOfLinearObjectiveCombinations = 0;
globalSetup->typeOfOptimizations = new OptimType[globalSetup->finalNoOfObjectives];
for(ii = 0; ii < globalSetup->finalNoOfObjectives; ii++) {
// read a line
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
// optimization type
if (strcmp("Min", pToken) == 0) {
globalSetup->typeOfOptimizations[ii] = Minimization;
}
else if (strcmp("Max", pToken) == 0) {
globalSetup->typeOfOptimizations[ii] = Maximization;
}
else {
fclose(fInput);
printf("Error! optimization type can either be Min or Max\n");
exit(1);
}
}
// constrained variables
// read the number of constrained variables
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
// the number can't be less than zero
if ((globalSetup->noOfRawConstraints = atoi(pToken)) < 0) {
fclose(fInput);
printf("Error! number of constraints should be >= 0\n");
exit(1);
}
else if (globalSetup->noOfRawConstraints == 0) {
if ((strlen(pToken) != 1) || (*pToken != '0')) {
fclose(fInput);
printf("Error! number of constraints should be >= 0\n");
exit(1);
}
}
}

```

```

globalSetup->noOfLinearConstraintCombinations = 0;
globalSetup->finalNoOfConstraints = globalSetup->noOfRawConstraints;
// penalty weights
globalSetup->penaltyWeights = new double[globalSetup->finalNoOfConstraints];
for(ii = 0; ii < globalSetup->finalNoOfConstraints; ii++) {
// read a line
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
globalSetup->penaltyWeights[iii] = atof(pToken);
}
// general parameters
// population size
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
if (strcmp("default", pToken) == 0) {
// Use a default value of 30*ell*log(ell);
printf("Using default population-sizing thumbrule: n = 30*ell*log(ell)\n");
// Take care of small problem sizes where log(ell) < 1
if (globalSetup->noOfDecisionVariables > 2)
globalSetup->populationSize = (int)(30*(globalSetup->noOfDecisionVariables)*log((double)(globalSetup->noOfDecisionVariables)));
else
globalSetup->populationSize = (int)(30*(globalSetup->noOfDecisionVariables));
//Round it to next nearest tenth number
if ((globalSetup->populationSize)%10) globalSetup->populationSize += (globalSetup->populationSize)%10;
printf("The population size used is: %d\n", globalSetup->populationSize);
}
// the number can't be less than or equal to zero
else if ((globalSetup->populationSize = atoi(pToken)) <= 0) {
fclose(fInput);
printf("The population size must be > 0\n");
exit(1);
}
else if ((globalSetup->populationSize%2) != 0) {
// the number can't be an odd number
fclose(fInput);
printf("Error! population size must be an even number\n");
exit(1);
}
// maximum generations
// the number can't be less than or equal to zero
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
if (strcmp("default", pToken) == 0) {
// Use a default value of 6*ell;
printf("Using default convergence-time thumbrule: tc = 6*ell\n");
globalSetup->maxGenerations = 6*(globalSetup->noOfDecisionVariables);
if ((globalSetup->maxGenerations)%10) globalSetup->maxGenerations += 10-(globalSetup->maxGenerations)%10;
printf("The maximum number of generations set is: %d\n", globalSetup->maxGenerations);
}
else if ((globalSetup->maxGenerations = atoi(pToken)) <= 0) {
fclose(fInput);
printf("Error! maximum number of generations must be > 0\n");
exit(1);
}
// replace proportion
// the number should be in (0.0, 1.0]
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
if (strcmp("default", pToken) == 0) {
// Setting a default value of 0.9
printf("Using default replacement proportion of 0.9\n");
globalSetup->replaceProportion = 0.9;
}

```

```

}
else if (((globalSetup->replaceProportion = atof(pToken)) <= 0.0) || (globalSetup->replaceProportion > 1.0)) {
fclose(fInput);
printf("Error! proportion of parent population that should be replaced must be > 0 and <= 1\n");
exit(1);
}
// niching (multimodal handling)
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
if (strcmp("default", pToken) == 0) {
// Using NoNiching by default
printf("No niching method is used by default\n");
globalSetup->nichingType = NoNiching;
}
// niching type
else if (strcmp("NoNiching", pToken) == 0) {
globalSetup->nichingType = NoNiching; globalSetup->nichingParameters = NULL;
}
else if (strcmp("Sharing", pToken) == 0) {
globalSetup->nichingType = Sharing;
}
else if (strcmp("RTS", pToken) == 0) {
globalSetup->nichingType = RTS;
}
else if (strcmp("DeterministicCrowding", pToken) == 0) {
globalSetup->nichingType = DeterministicCrowding;
}
else {
fclose(fInput);
printf("Error! valid niching types are: NoNiching, Sharing, RTS, and DeterministicCrowding\n");
exit(1);
}
// check niching type
if ((globalSetup->gaType == NSGA) && (globalSetup->nichingType != NoNiching)) {
fclose(fInput);
printf("Error! valid choice for niching types with NSGA is: NoNiching\n");
exit(1);
}
// read niching parameters
switch(globalSetup->nichingType) {
case NoNiching:
case DeterministicCrowding:
// no extra parameters
break;
case Sharing:
{
globalSetup->nichingParameters = new double[2];
if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {
printf("Using default sharing radius of 4.24\n");
((double *)globalSetup->nichingParameters)[0] = 4.24;
}
else
((double *)globalSetup->nichingParameters)[0] = atof(pToken);
if (((double *)globalSetup->nichingParameters)[0] <= 0.0) {
fclose(fInput);
printf("Error! niching radius must be > 0\n");
exit(1);
}
if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {
printf("Using default scaling of 1 for fitness sharing\n");
((double *)globalSetup->nichingParameters)[1] = 1.0;
}
else
((double *)globalSetup->nichingParameters)[1] = atof(pToken);
}
break;
case RTS:
{
globalSetup->nichingParameters = new int[1];
if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {
printf("Setting the window size for RTS using default rule: w = min(ell, n/20)\n");
}
}
}
}

```

```

if(globalSetup->noOfDecisionVariables < (globalSetup->populationSize/20))
((int*)(globalSetup->nichingParameters))[0] = globalSetup->noOfDecisionVariables;
else
((int*)(globalSetup->nichingParameters))[0] = (globalSetup->populationSize)/20;
//Adjust for small problem sizes: w = n/20;
if(globalSetup->noOfDecisionVariables < 20)
((int*)(globalSetup->nichingParameters))[0] = (globalSetup->populationSize)/20;
//Check if the window size is greater than the population size
if(((int*)globalSetup->nichingParameters)[0] > globalSetup->populationSize)
((int*)(globalSetup->nichingParameters))[0] = globalSetup->populationSize;
printf("The window size used for RTR is:%d\n", ((int*)globalSetup->nichingParameters)[0]);
}
else
((int*)(globalSetup->nichingParameters))[0] = atoi(pToken);
// the window size should be in (0, populationSize)
if (((int*)globalSetup->nichingParameters)[0] <= 0 || ((int*)globalSetup->nichingParameters)[0] > globalSetup->populationSize) {
fclose(fInput);
printf("Error! window size for RTR should be > 0 and less than the population size\n");
exit(1);
}
}
break;
default:
{
fclose(fInput);
printf("Error! valid choices for niching type are: NoNiching, Sharing, RTS, and DeterministicCrowding\n");
exit(1);
}
break;
}
// selection
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
if (strcmp("default", pToken) == 0) {
printf("Using tournament selection w/o replacement as a default selection method\n");
globalSetup->selectionType = TournamentWOR;
}
// selection type
else if (strcmp("TournamentWOR", pToken) == 0) {
globalSetup->selectionType = TournamentWOR;
}
else if (strcmp("SUS", pToken) == 0) {
globalSetup->selectionType = SUS;
}
else if (strcmp("Truncation", pToken) == 0) {
globalSetup->selectionType = Truncation;
}
else if (strcmp("RouletteWheel", pToken) == 0) {
globalSetup->selectionType = RouletteWheel;
}
else if (strcmp("TournamentWR", pToken) == 0) {
globalSetup->selectionType = TournamentWR;
}
else {
fclose(fInput);
printf("Error! valid selection methods are: RouletteWheel, SUS, TournamentWOR, TournamentWR, and Truncation\n");
exit(1);
}
// check selection type
if ((globalSetup->gaType == NSGA) && ((globalSetup->selectionType == SUS) || (globalSetup->selectionType == RouletteWheel))) {
fclose(fInput);
printf("Error! with NSGA, valid selection methods are: TournamentWOR, TournamentWR, and Truncation\n");
exit(1);
}
// read selection parameters
switch(globalSetup->selectionType) {
case TournamentWOR:
case Truncation:
case TournamentWR:
{
globalSetup->selectionParameters = new int[1];

```

```

if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {
printf("Using default tournament size of 2\n");
((int*)globalSetup->selectionParameters)[0] = 2;
}
else
((int*)globalSetup->selectionParameters)[0] = atoi(pToken);
}
break;
case SUS:
case RouletteWheel:
// no extra parameters
break;
default:
{
fclose(fInput);
printf("Error! invalid selection parameter\n");
exit(1);
}
break;
}
// Crossover
// crossover probability
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
if (strcmp("default", pToken) == 0) {
printf("Using a default crossover probability of 0.9\n");
globalSetup->xOverProbability = 0.9;
}
// the number should be [0.0, 1.0]
else if (((globalSetup->xOverProbability = atof(pToken)) < 0.0) || (globalSetup->xOverProbability > 1.0)) {
fclose(fInput);
printf("Error! crossover probability must be >= 0.0 and <= 1.0\n");
exit(1);
}
// crossover type
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
if (strcmp("default", pToken) == 0) {
printf("Using SBX as the default crossover method. Note that this might be an inappropriate choice if your variables are binary\n");
globalSetup->xOverType = SBX;
}
else if (strcmp("OnePoint", pToken) == 0) {
globalSetup->xOverType = OnePoint;
}
else if (strcmp("TwoPoint", pToken) == 0) {
globalSetup->xOverType = TwoPoint;
}
else if (strcmp("Uniform", pToken) == 0) {
globalSetup->xOverType = Uniform;
}
else if (strcmp("SBX", pToken) == 0) {
globalSetup->xOverType = SBX;
}
else {
fclose(fInput);
printf("Error! valid crossover types are: OnePoint, TwoPoint, Uniform, and SBX\n");
exit(1);
}
// read crossover parameters
switch(globalSetup->xOverType) {
case OnePoint:
case TwoPoint:
//no extra parameters
break;
case Uniform:
{
globalSetup->xOverParameters = new double[1];
if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {

```

```

printf("Using default genewise swap probability of 0.5\n");
((double*)globalSetup->xOverParameters)[0] = 0.5;
}
else
((double*)globalSetup->xOverParameters)[0] = atof(pToken);
if((((double*)globalSetup->xOverParameters)[0] <= 0.0) || (((double*)globalSetup->xOverParameters)[0] >= 1.0)) {
fclose(fInput);
printf("Genewise probability must be > 0.0 and < 1.0\n");
exit(1);
}
}
case SBX:
{
globalSetup->xOverParameters = new double[2];
if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {
printf("Using default genewise swap probability of 0.5\n");
((double*)globalSetup->xOverParameters)[0] = 0.5;
}
else
((double*)globalSetup->xOverParameters)[0] = atof(pToken);
if((((double*)globalSetup->xOverParameters)[0] <= 0.0) || (((double*)globalSetup->xOverParameters)[0] >= 1.0)) {
fclose(fInput);
printf("Error! genewise probability must be > 0.0 and < 1.0\n");
exit(1);
}
if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {
printf("Using default polynomial order for SBX: 10\n");
((double*)globalSetup->xOverParameters)[1] = 10;
}
else
((double*)globalSetup->xOverParameters)[1] = atof(pToken);
if(((double*)globalSetup->xOverParameters)[1] < 0.0) {
fclose(fInput);
printf("Error! genewise probability must be >= 0.0\n");
exit(1);
}
}
break;
default:
{
fclose(fInput);
printf("Error! invalid crossover parameter\n");
exit(1);
}
break;
}
// mutation
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
if (strcmp("default", pToken) == 0) {
printf("Using a default mutation probability of 0.1\n");
globalSetup->mutationProbability = 0.1;
}
// mutation probability
// the number should be [0.0, 1.0]
else if (((globalSetup->mutationProbability = atof(pToken)) < 0.0) || (globalSetup->mutationProbability > 1.0)) {
fclose(fInput);
printf("Error! mutation probability must be >= 0.0 and <= 1.0.\n");
exit(1);
}
// mutation type
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
if (strcmp("default", pToken) == 0) {
printf("Using Polynomial as the default mutation method. Note that this might be an inappropriate choice if your variables are binary\n");
globalSetup->mutationType = Polynomial;
}
else if (strcmp("Selective", pToken) == 0) {

```

```

globalSetup->mutationType = Selective;
}
else if (strcmp("Genewise", pToken) == 0) {
globalSetup->mutationType = Genewise;
}
else if (strcmp("Polynomial", pToken) == 0) {
globalSetup->mutationType = Polynomial;
}
else {
fclose(fInput);
printf("Error! valid mutation types are: Selective, Genewise, and Polynomial\n");
exit(1);
}
// read mutation parameters
switch(globalSetup->mutationType) {
case Selective:
// no extra parameters
break;
case Polynomial:
{
globalSetup->mutationParameters = new int[1];
if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {
printf("Using a default value for the polynomial probability: 20\n");
((int*)globalSetup->mutationParameters)[0] = 20;
}
else
((int*)globalSetup->mutationParameters)[0] = atoi(pToken);
if(((int*)globalSetup->mutationParameters)[0] < 0) {
fclose(fInput);
printf("Error! polynomial order for polynomial mutation must be > 0\n");
exit(1);
}
}
break;
case Genewise:
{
globalSetup->mutationParameters = new double[globalSetup->noOfDecisionVariables];
for (ii = 0 ; ii < globalSetup->noOfDecisionVariables ; ii++) {
if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {
printf("Using default std. deviation of 10% of the variable range\n");
((double*)globalSetup->mutationParameters)[ii] = 0.1*(globalSetup->variableRanges[ii][1] - globalSetup->variableRanges[ii][0]);
}
else
((double*)globalSetup->mutationParameters)[ii] = atof(pToken);
if(((double*)globalSetup->mutationParameters)[ii] <= 0.0) {
fclose(fInput);
printf("Error! standard deviation for gene%d for genewise mutation must be > 0\n", ii);
exit(1);
}
}
break;
default:
{
fclose(fInput);
printf("Error! invalid mutation parameter\n");
exit(1);
}
break;
}
// scaling method
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
if (strcmp("default", pToken) == 0) {
printf("Scaling is not used by default\n");
globalSetup->scalingMethod = NoScaling;
}
// scaling method
else if (strcmp("NoScaling", pToken) == 0) {
globalSetup->scalingMethod = NoScaling;
}
}

```

```

else if (strcmp("Ranking", pToken) == 0) {
globalSetup->scalingMethod = Ranking;
}
else if (strcmp("SigmaScaling", pToken) == 0) {
globalSetup->scalingMethod = SigmaScaling;
}
else {
fclose(fInput);
printf("Error! valid scaling methods are: NoScaling, Ranking, and SigmaScaling\n");
exit(1);
}
// read scaling parameters
switch(globalSetup->scalingMethod) {
case NoScaling:
case Ranking:
// no extra parameters
break;
case SigmaScaling:
{
globalSetup->scalingParameters = new double[1];
if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {
((double*)globalSetup->scalingParameters)[0] = 1.0;
}
else
((double*)globalSetup->scalingParameters)[0] = atof(pToken);
if (((double*)globalSetup->scalingParameters)[0] <= 0.0) {
fclose(fInput);
printf("Error! scaling parameter for SigmaScaling must be > 0\n");
exit(1);
}
}
break;
default:
{
fclose(fInput);
printf("Error! invalid scaling parameter\n");
exit(1);
}
break;
}
// constraint method
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
if (strcmp("default", pToken) == 0) {
if (globalSetup->finalNoOfConstraints == 0) {
printf("Using no constraint handling method by default\n");
globalSetup->constraintMethod = NoConstraints;
}
else {
printf("Using tournament selection as the default constraint handling method\n");
globalSetup->constraintMethod = Tournament;
}
}
// constraint method
else if (strcmp("NoConstraints", pToken) == 0) {
globalSetup->constraintMethod = NoConstraints;
}
else if (strcmp("Penalty", pToken) == 0) {
globalSetup->constraintMethod = Penalty;
}
else if (strcmp("Tournament", pToken) == 0) {
globalSetup->constraintMethod = Tournament;
}
else {
fclose(fInput);
printf("Error! valid constraint handling methods are: NoConstraint, Penalty, and Tournament\n");
exit(1);
}
// check constraint method
if ((globalSetup->gaType == NSGA) && (globalSetup->constraintMethod == Penalty)) {
fclose(fInput);
}

```

```

printf("Error! penalty based constraint handling method cannot be used with NSGA\n");
exit(1);
}
if ((globalSetup->finalNoOfConstraints == 0) && (globalSetup->constraintMethod != NoConstraints)) {
fclose(fInput);
printf("Error! valid constraint-handling method when there are no constraints is NoConstraints\n");
exit(1);
}
// read penalty function
switch(globalSetup->constraintMethod) {
case NoConstraints:
case Tournament:
// no extra parameters
break;
case Penalty:
{
// penalty function
if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {
printf("Using linear penalty be default\n");
globalSetup->penaltyFunction = Linear;
}
else if (strcmp("Linear", pToken) == 0) {
globalSetup->penaltyFunction = Linear;
}
else if (strcmp("Quadratic", pToken) == 0) {
globalSetup->penaltyFunction = Quadratic;
}
else {
fclose(fInput);
printf("Error! valid penalty function methods are: Linear and Quadratic\n");
exit(1);
}
break;
default:
{
fclose(fInput);
printf("Error! invalid constraint-handling method parameter\n");
exit(1);
}
break;
}
// local search method
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
if (strcmp("default", pToken) == 0)
{
printf("Using no local search by default\n");
globalSetup->localSearchMethod = NoLocalSearch;
}
// local search method
else if (strcmp("NoLocalSearch", pToken) == 0) {
globalSetup->localSearchMethod = NoLocalSearch;
}
else if (strcmp("SimplexSearch", pToken) == 0) {
globalSetup->localSearchMethod = SimplexSearch;
}
else {
fclose(fInput);
printf("Error! valid local search methods are: NoLocalSearch and SimplexSearch\n");
exit(1);
}
// check local search method
if ((globalSetup->localSearchMethod != NoLocalSearch) && (globalSetup->gaType == NSGA)) {
fclose(fInput);
printf("Error! cannot use local search with NSGA\n");
exit(1);
}
switch(globalSetup->localSearchMethod) {
case NoLocalSearch:
// no extra parameters

```

```

break;
case SimplexSearch:
if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {
printf("Using default tolerance of 0.001\n");
globalSetup->maxLocalTolerance = 1.0E-3;
}
else
globalSetup->maxLocalTolerance = atof(pToken);
if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {
printf("Setting maximum local search evaluations to 20\n");
globalSetup->maxLocalEvaluations = 20;
}
else
globalSetup->maxLocalEvaluations = atoi(pToken);
if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {
printf("Using default local penalty parameter of 1.0\n");
globalSetup->initialLocalPenaltyParameter = 1.0;
}
else
globalSetup->initialLocalPenaltyParameter = atof(pToken);
if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {
printf("Using default local update parameter of 2.0\n");
globalSetup->localUpdateParameter = 2.0;
}
else
globalSetup->localUpdateParameter = atof(pToken);
if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {
printf("Using default Lamarckian probability of 0.15\n");
globalSetup->lamarckianProbability = 0.15;
}
else
globalSetup->lamarckianProbability = atof(pToken);
if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {
printf("Using default overall local search probability of 0.5\n");
globalSetup->localSearchProbability = 0.5;
}
else
globalSetup->localSearchProbability = atof(pToken);
break;
default:
{
fclose(fInput);
printf("Error! invalid local search parameters\n");
exit(1);
}
break;
}
// stopping criteria
// noOfStoppingCriterias
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
if (strcmp("default", pToken) == 0)
{
printf("Using no extra stopping criterias by default.\n");
globalSetup->noOfStoppingCriterias = 0;
}
// the number can't be less than zero
else if ((globalSetup->noOfStoppingCriterias = atoi(pToken)) < 0) {
fclose(fInput);
printf("Error! number of stopping criterias must be > 0\n");
exit(1);
}
else if (globalSetup->noOfStoppingCriterias == 0) {
if ((strlen(pToken) != 1) || (*pToken != '0')) {
fclose(fInput);
printf("Error! number of stopping criterias must be a number!\n");
exit(1);
}
}
// allocate memory
if (globalSetup->noOfStoppingCriterias > 0) {

```

```

// number of generation window
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
printf("The default window size used in stopping critiria is: 5.\n");
globalSetup->genNumWindow = 5;
}
else
globalSetup->genNumWindow = atoi(pToken);
// the number should be greater than zero
if (globalSetup->genNumWindow <= 0) {
fclose(fInput);
printf("Error! window size used in stopping critieria must be > 0\n");
exit(1);
}
globalSetup->otherStoppingCriteria = new StoppingCriterias [globalSetup->noOfStoppingCriterias];
globalSetup->stoppingParameter = new double [globalSetup->noOfStoppingCriterias];
for (ii = 0 ; ii < globalSetup->noOfStoppingCriterias ; ii++) {
// read a line
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, Please see the documentation.\n");
exit(1);
}
// stopping criterion type
if (strcmp("NoOfEvaluations", pToken) == 0) {
globalSetup->otherStoppingCriteria[ii] = NoOfEvaluations;
}
else if (strcmp("FitnessVariance", pToken) == 0) {
globalSetup->otherStoppingCriteria[ii] = FitnessVariance;
}
else if (strcmp("AverageFitness", pToken) == 0) {
globalSetup->otherStoppingCriteria[ii] = AverageFitness;
}
else if (strcmp("AverageObjective", pToken) == 0) {
globalSetup->otherStoppingCriteria[ii] = AverageObjective;
}
else if (strcmp("ChangeInBestFitness", pToken) == 0) {
globalSetup->otherStoppingCriteria[ii] = ChangeInBestFitness;
}
else if (strcmp("ChangeInAvgFitness", pToken) == 0) {
globalSetup->otherStoppingCriteria[ii] = ChangeInAvgFitness;
}
else if (strcmp("ChangeInFitnessVar", pToken) == 0) {
globalSetup->otherStoppingCriteria[ii] = ChangeInFitnessVar;
}
else if (strcmp("ChangeInBestObjective", pToken) == 0) {
globalSetup->otherStoppingCriteria[ii] = ChangeInBestObjective;
}
else if (strcmp("ChangeInAvgObjective", pToken) == 0) {
globalSetup->otherStoppingCriteria[ii] = ChangeInAvgObjective;
}
else if (strcmp("NoOfFronts", pToken) == 0) {
globalSetup->otherStoppingCriteria[ii] = NoOfFronts;
}
else if (strcmp("NoOfGuysInFirstFront", pToken) == 0) {
globalSetup->otherStoppingCriteria[ii] = NoOfGuysInFirstFront;
}
else if (strcmp("ChangeInNoOfFronts", pToken) == 0) {
globalSetup->otherStoppingCriteria[ii] = ChangeInNoOfFronts;
}
else if (strcmp("BestFitness", pToken) == 0) {
globalSetup->otherStoppingCriteria[ii] = BestFitness;
}
else {
fclose(fInput);
printf("Error! invalid stopping criteria parameter\n");
exit(1);
}
// check criteria
switch(globalSetup->otherStoppingCriteria[ii]) {
case NoOfEvaluations:
case FitnessVariance:
// 1 - nothing to check
break;
case AverageFitness:

```

```

case AverageObjective:
case ChangeInBestFitness:
case ChangeInAvgFitness:
case ChangeInFitnessVar:
case ChangeInBestObjective:
case ChangeInAvgObjective:
case BestFitness:
// 2 - SGA w/o MM only
if ((globalSetup->gaType != SGA) || (globalSetup->nichingType != NoNiching)) {
fclose(fInput);
printf("Cannot use the following stopping critier with NSGA or when Niching is used:\n");
printf("\t AverageFitness\n\t AverageObjective\n");
printf("\t ChangeInBestFitness\n\t ChangeInAvgFitness\n");
printf("\t ChangeInFitnessVar\n\t ChangeInBestObjective\n");
printf("\t ChangeInAvgObjective\n\t BestFitness\n");
printf("Error! invalid choice of stopping criteria");
exit(1);
}
break;
case NoOfFronts:
case NoOfGuysInFirstFront:
case ChangeInNoOfFronts:
// 3 - NSGA only
if (globalSetup->gaType != NSGA) {
fclose(fInput);
printf("Error! following stopping criteria can be used only with NSGA:\n");
printf("\t NoOfFronts\n\t NoOfGuysInFirstFront\n\t ChangeInNoOfFronts\n");
exit(1);
}
break;
default:
{
fclose(fInput);
printf("Error! invalid stopping criteria\n");
exit(1);
}
break;
}
// parameter
if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {
fclose(fInput);
printf("Error! invalid stopping criteria parameter\n");
exit(1);
}
((double*)globalSetup->stoppingParameter)[ii] = atof(pToken);
}
}
// Initial population generation
// Load population
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
if (strcmp("default", pToken) == 0)
{
printf("Using random initialization by default.\n");
globalSetup->loadPopulation = false;
globalSetup->evaluateAgain = true;
}
// Check if the load population is 0 or 1
else {
// load the population from a file
if (atoi(pToken) == 1) {
globalSetup->loadPopulation = true;
globalSetup->populationFileName = new char[256];
// Read the name of the file to load the population from
if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {
fclose(fInput);
printf("Error! invalid file name to load the population from, see documentation for more details.\n");
exit(1);
}
strcpy(globalSetup->populationFileName, pToken);
// Evaluate the population or not

```

```

if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {
printf("Evaluating the initial population by default.\n");
globalSetup->evaluateAgain = true;
}
else if (atoi(pToken) == 0) {
globalSetup->evaluateAgain = false;
}
else
globalSetup->evaluateAgain = true;
}
// Use random initialization
else {
globalSetup->loadPopulation = false;
globalSetup->evaluateAgain = true;
}
}
// Save population
if ((pToken = readOneLine(caBuf, ciBufSize, fInput)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
if (strcmp("default", pToken) == 0) {
printf("Saving the evaluated individuals to a file by default.\n");
globalSetup->savePopulation = true;
globalSetup->saveEvalSolutions = new char[256];
printf("Enter the filename you want to save the population to.\n");
fflush(stdout);
scanf("%s", globalSetup->saveEvalSolutions);
}
// Check if the save population is 0 or 1
else {
fflush(stdout);
// Save the population to a file
if (atoi(pToken) == 1) {
globalSetup->savePopulation = true;
// Filename to save the evaluated solutions to.
globalSetup->saveEvalSolutions = new char[256];
if ((pToken = strtok(NULL, BLANK_STR)) == NULL) {
fclose(fInput);
printf("Error in the input file, please refer to the documentation\n");
exit(1);
}
strcpy(globalSetup->saveEvalSolutions, pToken);
}
}
fclose(fInput);
fflush(stdout);
if (globalSetup->savePopulation) {
fOutput = fopen(globalSetup->saveEvalSolutions, "w");
fprintf(fOutput, "%s\n");
for(ii = 0; ii < globalSetup->noOfDecisionVariables; ii++) {
fprintf(fOutput, "var #%d\t", ii);
}
for(ii = 0; ii < globalSetup->finalNoOfObjectives; ii++) {
fprintf(fOutput, "obj #%d\t", ii);
}
if (globalSetup->finalNoOfConstraints > 0) {
for(ii = 0; ii < globalSetup->finalNoOfConstraints; ii++) {
fprintf(fOutput, "const #%d\t", ii);
}
fprintf(fOutput, "penalty");
}
fprintf(fOutput, "\n");
fclose(fOutput);
}
GA ga;
if (globalSetup->gaType == SGA) {
while(ga.generate());
}
else {
while(ga.nsgaGenerate());
}
delete [] (globalSetup->penaltyWeights);

```

```

delete [] (globalSetup->variableTypes);
for(ii = 0; ii < globalSetup->noOfDecisionVariables; ii++) {
delete [] (globalSetup->variableRanges[ii]);
}
delete [] (globalSetup->variableRanges);
delete [] (globalSetup->typeOfOptimizations);
if((globalSetup->selectionType == TournamentWR) || (globalSetup->selectionType == TournamentWOR) || (globalSetup->selectionType == Truncation)) {
delete [] (int*) (globalSetup->selectionParameters);
}
if((globalSetup->xOverType==Uniform) || (globalSetup->xOverType == SBX))
delete [] (double*) (globalSetup->xOverParameters);
if (globalSetup->mutationType == Polynomial)
delete [] (int*) (globalSetup->mutationParameters);
else if (globalSetup->mutationType == GeneWise)
delete [] (double*) (globalSetup->mutationParameters);
if (globalSetup->nichingType == RTS)
delete [] (int*) (globalSetup->nichingParameters);
if (globalSetup->nichingType == Sharing)
delete [] (double*) (globalSetup->nichingParameters);
if (globalSetup->scalingMethod == SigmaScaling)
delete [] (double*) (globalSetup->scalingParameters);
if (globalSetup->noOfStoppingCriteria > 0) {
delete [] globalSetup->otherStoppingCriteria;
delete [] (double*) (globalSetup->stoppingParameter);
}
if (globalSetup->loadPopulation)
delete [] (globalSetup->populationFileName);
if (globalSetup->savePopulation)
delete [] (globalSetup->saveEvalSolutions);
return 1;
}

/*****mean_draw.sh*****/
#!/bin/sh
awk '{for(i=1;i<=NF;i++) {s+=$i; mean1=s/8877}} END {print mean1}' draw_lay2_step1_per2.dat
awk '{for(i=1;i<=NF;i++) {s+=$i; mean2=s/8877}} END {print mean2}' draw_lay2_step2_per2.dat
awk '{for(i=1;i<=NF;i++) {s+=$i; mean3=s/8877}} END {print mean3}' draw_lay2_step3_per2.dat
awk '{for(i=1;i<=NF;i++) {s+=$i; mean4=s/8877}} END {print mean4}' draw_lay2_step4_per2.dat
awk '{for(i=1;i<=NF;i++) {s+=$i; mean5=s/8877}} END {print mean5}' draw_lay2_step5_per2.dat
awk '{for(i=1;i<=NF;i++) {s+=$i; mean6=s/8877}} END {print mean6}' draw_lay2_step6_per2.dat
awk '{for(i=1;i<=NF;i++) {s+=$i; mean7=s/8877}} END {print mean7}' draw_lay2_step7_per2.dat
awk '{for(i=1;i<=NF;i++) {s+=$i; mean8=s/8877}} END {print mean8}' draw_lay2_step8_per2.dat
awk '{for(i=1;i<=NF;i++) {s+=$i; mean9=s/8877}} END {print mean9}' draw_lay2_step9_per2.dat
exit

/*****read_wel.sh*****/
#!/bin/bash awk '$1==2 {print $1,$2,$3}' ULTIMO_2.wel > data.wel
awk '$1==2 {print $4}' ULTIMO_2.wel > bombeo.dat
exit

/*****sum_pump.sh*****/
#!/bin/sh awk '{if (NR>1098 && $1>10 && $1<10000000) print}' output.dat > sum_pump.dat

/*****write_output.sh*****/
#!/bin/sh awk '{ {NR>1} for(i=1;i<=NF;i++) { print($i) } }' evaluatedSolutions.txt > output.dat exit

```


Bibliografía

- [1] Evaluación mundial sobre abastecimiento de agua y saneamiento. Technical report, OMS/UNICEF, 2000.
- [2] *PEST, model-independent parameter estimation*. Watermark Numerical Computation, Townville, 2000.
- [3] N. Andrei. *Advanced Mathematical Programming. Theory, Computational methods, Applications*. Technical Press, Bucharest, 1999.
- [4] N. Andrei. *Mathematical Programming. Interior point methods*. Technical Press, Bucharest, 1999.
- [5] N. Andrei. *Semidefinite Programming*. MatrixRom, Bucharest., 2001.
- [6] N. Andrei. *Convergence of optimization algorithms*. Technical Press, Bucharest, 2004.
- [7] N. Andrei. *Theory versus Empiricism in Analysis of Optimization Algorithms*. Technical Press, Bucharest., 2004.
- [8] T. Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1995.
- [9] R. Bellman. *Dynamic Programming*. Princeton Univ. Press., New Jersey, 1957.
- [10] Ferris M. y Ionnidis Y. Bennet, k. *Proceedings of the fourth international conference on genetic algorithms.*, chapter A genetic algorithm for ata base query optimization., pages 400–407. Morgan Kaufmann Plublishers, San Mateo, Cal., 1991.
- [11] S. Bennet. *A history of control engineering 1800-1930*, volume 8 of *IEE Control Engineering Series*. Peter Peregrinus Ltd, London, 1979.
- [12] V. Bithin, D. y Harikrishna. *Optimization applications in water resources systems engineering*, 2004.

- [13] H. Black. Stabilized feedback amplifiers. Technical journal, Bell Systems, 1934.
- [14] Ghaoui L.E. Feron E. y Balakrishnan V. Boyd, S. Linear matrix inequalities in system and control theory. *SIAM*, 1994.
- [15] H. Brenermann. *The evolution of intelligence. The nervous system as a model of its environment*. University of Washington, Seattle, 1958.
- [16] H. Brenermann. *Self-Organizing Systems*, chapter Optimization through evolution and recombination., pages 93–106. Spartan Books., Washington D.C., 1962.
- [17] H. Brenermann. *Cybernetic Problems in Bionics*, chapter Numerical optimization procedures derived from biological evolution processes, pages 543–561. Gordon and Breach,, New York, 1968.
- [18] M. Brenermann, H. y Rogson. *An evolution-type search method for convex sets*. ONR, Berkley, California, 1964.
- [19] M. Brenermann, H. y Rogson. *Natural Automata and Useful Simulations*, chapter Global properties of evolution processes., pages 3–41. Spartan Books, Wasington D.C., 1996.
- [20] Burago M y Friend A. Bressan, A. Blocking strategies for a fire control problem. 2008.
- [21] M. Burke J. y Monch. Groundwater in urban development: Assesing management needs and formulating oily strategies. Technical Report 390, Banco Mundial, 1997.
- [22] M. Burke J. y Monch. Groundwater ans society: resources, tensions and opportunities. *Unated Nations Publication ST/ESA/205*, 2000.
- [23] W. Cannon. *The Wisdom of the Body*. Norton and Company, New York, 1932.
- [24] R. Caravias, J. y Landa. *Agua, mediambiente y sociedad*. UNAM, COLMEX y Fundación Río Arrorte., 1^o edition, 2005.
- [25] A Carrillo. Aplicaciones del Cómputo en Paralelo a la Modelación de Sistemas Terrestres, 2006.
- [26] Shoemaker C. y Liu P. Chang, L. Optimal time-varying pumping rates for groundwater remediation: application of a constrained optimal control algorithm. *Water Resour. Res.*, 28:3157–3173, 1992.
- [27] C. Darwin. *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. 1859.

- [28] B. Das, A. y Datta. Development of multiobjective management models for coastal aquifers. *J. Water Res. Planning Manage., Am. Soc. Civ. Eng.*, 125:76–78, 1999a.
- [29] B. Das, A. y Datta. Development of management models for sustainable use of coastal aquifers. *J. Irrigation Drainage Eng., Am. Soc. Civ. Eng.*, 125:112–121, 1999b.
- [30] B. Das, A. y Datta. Optimisation based solution of density dependent seawater intrusion in coastal aquifers. *J. Hydrol. Eng., Am. Soc. Civ. Eng.*, 5:82–89, 2000.
- [31] L. Davis, editor. *Genetic algorithms and Simulated Annealing*. Morhan Kaufmann Publishers, San Mateo, Cal., 1987.
- [32] Gupta V. y Sorooshian S. Duan, Q. Effective and efficient global optimization for conceptual rainfall-runoff models. *Water Resour. Res.*, 28:1015–1031, 1992.
- [33] L. Evans. *An Introduction to Mathematical Optimal Control Theory Version 0.2*.
- [34] D. Fogel. *Evolutionary computation: toward a new Philosophy of machine intelligence*. IEEE, Piscataway, NJ, 1995.
- [35] A. Fraser. Simulation of genetic systems by automatic digital computers i. introduction., *Australian Journal of Biological Sciences*, 10:484–491, 1957.
- [36] A. Fraser. Simulation of genetic systems by automatic digital computers ii. effects of linkage on rates of advance under selection. *Australian Journal of Biological Sciences*, 10:150–162, 1957.
- [37] A. Fraser. Simulation of genetic systems by automatic digital computers vi. epistasis. *Australian Journal of Biological Sciences*, 13:150–162, 1960.
- [38] A. Fraser. *Computer Models in Genetics*,. Mc. Graw Hill, New York, 1970.
- [39] R. Gardner, O. y Engelman. *Sustaining Water, Easing Scarcity: A second Update*. Population Action International, 1997.
- [40] A. Hall. *Application of Circuit Theory to the Design of Servomechanisms*. J. Franklin Inst., 1966.
- [41] R. Hall. *Governors and governing mechanisms*. The Technical Publishing Co., Manchester, 2° edition, 1907.
- [42] H. Hallaji K. y Yazicigil. Optimal management of a coastal aquifer in southern turkey. *J. Water Res. Planning Manage. Div., Am. Soc. Civ. Eng.*, 122:233–244, 1996.

- [43] G. Herrera, I y Pinder. *Fundamental Concepts in Mathematical Modelling of engineering and science systems*. 2009.
- [44] I. Herrera. *Modelación matemática de sistemas terrestres*. 2008.
- [45] M. Hill. *Methods and guidelines foreffective model calibration*. Technical Report 98-4005, USGS Water rosources investigations report, 1998.
- [46] J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [47] A. Hurwitz. On the conditions under which an equation has only roots with negative real parts. *Selected Papers on Mathematical Trends in Control Theory*, 1964.
- [48] G. Israel. On the contribution of volterra and lotka to the development of modern biomathematics. *History and philosophy of the life sciences*, 10(1):37–49, 1988.
- [49] Willis R. y Yeh W. Jones, L. Optimal control of nonlinear groundwater hydraulics using differential dynamic programming. *Water Resour. Res.*, 23:2097–2107, 1987.
- [50] R. Kalman. *Encyclopaedia Britannica*, chapter Optimization, mathematical theory of, control theory,, pages 636–638. 50° edition, 1974.
- [51] J. Lee, A. y Aronafsky. linear programming model for scheduling crude oil production. *J. Petrol Technol.*, 213:51–54, 1958.
- [52] F. Lewis. *Introduction to Modern Control Theory*. Prentice-Hall., 1992.
- [53] M. Lyapunov. *Probleme general de la stabilite du mouvement*, volume 17 of *Annals of Mathematics Studies*. Princeton University Press, Princeton, 1947.
- [54] T. Maddock. Algebraic technological function from a simulation model. *Water Resour. Res.*, 8:129–134, 1972a.
- [55] J. Maxwell. On governors. *Proceedings of the Royal Society of London*, 16:270–283, 1867.
- [56] O. Mayr. *The origins of feedback control*. MIT Press, Cambridge, MA., 1970.
- [57] W. McPhee, J. y Yeh. Multiobjective optimization for sustainable groundwater management in semiarid regions. eScholarship repository <http://repositories.cdlib.org/postprints/260>, 2004.
- [58] Z. Michalewicz. *Genetic Algorithms + Data structures = Evolution Programs*. Springer, 3° edition, 1995.

- [59] H. Nyquist. Regeneration theory. Technical journal, Bell Systems, 1932.
- [60] K. Ogata. *Ingeniería de Control Moderna*. Prentice Hall., 1° edition, 1980.
- [61] K. Ogata. *Dinámica de sistemas físicos*. Prentice Hall., 1° edition, 1987.
- [62] M. Poeter, E. y Hill. A computer code for inverse modelling. Technical Report 98-4080, USGS Water resources investigations report, 1998.
- [63] Boltyansky V. Gamkrelidze R. y Mishchenko E. (1962) Pontryagin, L. *The Mathematical Theory of Optimal Processes*. Wiley, New York, 1962.
- [64] j. Refsgaard and H. Henriksen. Modeling guidelines, terminology and guiding principles. *Adv. Water Resour.*, 27:71–84, 2004.
- [65] A.T. Routh, E.J. y Fuller. *Stability of motion*. Taylor & Francis, 1875.
- [66] E.J. Routh. *A Treatise on the Stability of a Given State of Motion, Particularly Steady Motion*. Macmillan and Co, 1877.
- [67] Lee A. Schoups, G. and M. Groelick. Multi-objective calibration of a surface water-groundwater model. *HESD*, 2:2061–2109, 2005.
- [68] E. Sontag. *Mathematical Control Theory*. Springer, 2° edition, 1998.
- [69] H. Sussmann. Re*sultats re*cents sur les courbes optimales. *Journe*e Annuelle de la Societé Mathe*matique de France.*, 2000.
- [70] A. Taylor, P y Blum. *Ecosystem as circuits: diagrams and the limits of physical analogies*. Biology and Philosophy. Springer Netherlands, 2006.
- [71] M. Turing. Computing machinery and intelligence. *Mind*, 59:94–101, 1950.
- [72] F. Viniegra. Mecánica de los medios deformables. 2007.
- [73] Gupta H. Boeten W. Vrugt, J. and S. Sorooshian. A shuffled complex evolution metropolis algorithm for optimization and uncertainty assessment of hydrological model parameters. *Water Resour. Res.*, 39:1201–1218, 2003.
- [74] I. Vyshnegradsky. *On Controllers of Direct Action*. Izv. SPB Tekhnolog. Inst, 1877.
- [75] N. Wiener. *Extrapolation. Interpolation, and Smoothing of Stationary time Series with Engineering Applications*. Wiley, New York, 1949.

- [76] Amlan D. y Bithin D. Application of optimisation techniques in groundwater quantity and quality management. *Sadhana*, 26:293–316, 2001.
- [77] J. Zektser, I. y Margat. Groundwater resources of the world and their use. 1997.
- [78] E. Zuazua. *Teoría matemática del control: motor del desarrollo científico, tecnológico y social*. Departamento de matemática aplicada de la Universidad Complutense de Madrid.

Índice alfabético

- acción, 35
- Algoritmos genéticos (AG), 63
- analogías, 19

- balance global ecuación, 50
- bang-bang, controles, 28

- calculo variacional, 35
- Calibración multiobjetivo, 69
- Computo evolutivo, 62
- condición de máximo, 38
- control, 15
- Control óptimo del agua subterránea, 44
- control óptimo, 26
- control óptimo, solución gráfica, 31
- controlabilidad, 31
- Criterio de Controlabilidad, 33

- Darcy, ley, 51
- derivada material, 49
- diagrama de bloques, 19
- Diferencias finitas, 54
- dual, variable, 33

- ecuaciones constitutivas, 46
- ecuaciones de balance, 46
- estados alcanzables, 31
- Euler-Lagrange, ecuaciones, 36
- euleriana representación, 49
- Evaluación adecuación, AG, 67

- fluctuaciones, 15

- Flujo subterráneo, ecuación de., 50
- fuerza generalizada, 37
- Función de Transferencia, 17
- funcional recompensa, 27

- gráfico de flujo, 22

- Hamilton, ecuaciones, 37
- Hamiltoniano, 37
- hipótesis del continuo, 46
- Historia control, 13

- incendios forestales, 41

- lagrangiana representación, 49
- lagrangiano, 35

- Manejo óptimo agua subterránea, 55
- Mason, formula de ganancias de, 23
- Medios deformables, 45
- momento generalizado, 37
- multiplicador de Lagrange, 38

- observabilidad, teorema, 34
- observabilidad, 33
- Operadores genéticos, AG, 68
- Optimización usando AG, un ejemplo, 65
- optimización, 16
- optimización, 24

- Pontryagin, Principio Maximal, 39
- Programación dinámica (PD), 59
- Programación lineal (PL), 56

Programación no lineal (PNL), 57

Propiedades intensivas y extensivas, 49

recompensa, 27

Representación genética, AG, 66

retroalimentación, 15

sistema, 15

Solución numérica, flujo subterráneo., 53

teoría clásica de control, 16

teoría de control óptimo, 24