

Modelo de multicapas para la rehabilitación de sistemas de drenaje urbano: optimización multiobjetivo y computación en paralelo

A multi-tier model for the rehabilitation of urban drainage systems: multi-objective optimization and parallel computing

W. Barreto^{1*} y Y. Garcia-Orellana^{2*}

¹Departamento de Hidráulica. Decanato de Ing. Civil. Universidad Centroccidental "Lisandro Alvarado". Barquisimeto, Venezuela.
*wbarreto@ucla.edu.ve

²Departamento de Ingeniería. Decanato de Agronomía. Universidad Centroccidental "Lisandro Alvarado". Cabudare, Venezuela.
*yelitzagarcia@garcia.edu.ve

Fecha de recepción del artículo: 20/08/2012 Fecha de aceptación del artículo: 28/12/2012

Resumen

Este trabajo introduce un modelo multicapas para la Rehabilitación de Sistemas de Drenaje Urbano (RSDU). Un algoritmo de optimización multiobjetivo, utilizando algoritmos genéticos y empleando computación en paralelo, fue desarrollado para enfrentar problemas que son costosos en tiempo computacional. El estudio se plantea como un problema de optimización limitado (debido a la escasez de recursos) en un contexto de decisión multicriterio, involucrando varias partes en conflicto con diferentes intereses. El enfoque de "multicapas" consiste en usar varios niveles que corresponden a diferentes dimensiones (o sub-problemas) del problema total, efectuándose una evaluación del tiempo de ejecución basada en el número de procesadores. Los resultados muestran una prometedora y económica alternativa para problemas computacionalmente costosos, como es el drenaje urbano, usando optimización multi-objetivos. El tiempo de ejecución total se redujo en más del 60% de 12,8 a 4,8 horas, por lo que el método puede ser aplicado en países en desarrollo y en pequeñas oficinas de consultoría.

Palabras claves

Drenaje urbano, multiobjetivo, optimización, computación paralelo.

Abstract

This paper presents the development of a parallel multi-tier approach to address the problem of optimising the Rehabilitation of Urban Drainage Systems (UDS). A parallel multi-objective GA-based algorithm has been developed to deal with problems that are expensive in computational time. The problem is posed as a constrained optimization problem (due to limited resources) in a multi-criteria context (due to many stakeholders with conflicting interests). A "multi-tier" approach is introduced, where different tiers correspond to different dimensions (or sub-problems) of the whole problem. An evaluation of the performance time is carried out based on the number of computers or processors. The first results show a promising and cheap solution for "expensive tasks" like urban drainage optimization in the multi-criteria context, it was possible to reduce the total running time from 12.8 to 4.8 hours for the test problem.

Keywords

Urban drainage; multi-objective optimization; parallel computing.

Introducción

Los sistemas de drenaje urbano (SDUs) se degradan con el tiempo debido a múltiples factores

como el envejecimiento de las tuberías, el crecimiento poblacional, los cambios climáticos, etc. La rehabilitación de este tipo de sistemas es una tarea muy compleja, con participación de un gran número de variables, consideraciones y limitaciones. Los encargados de adoptar decisiones tienen que invertir un presupuesto, usualmente limitado, tratando de maximizar los beneficios por cantidad monetaria invertida. Para ello, se deben desarrollar herramientas de optimización que sean eficientes en convergencia y tiempo computacional.

La rehabilitación de los SDUs es un proceso que involucra múltiples criterios, requiriéndose del uso de enfoques multiobjetivo. Sin embargo, debido a la complejidad de dichos sistemas, varios problemas deben ser abordados durante este proceso, por ejemplo, la selección del modelo hidráulico es una cuestión importante. Los modelos de una dimensión son utilizados con frecuencia; estos se comportan bien para las tuberías enterradas del sistema mas no así para el sistema de drenaje superficial, donde lo que domina realmente es el comportamiento en dos dimensiones para la determinación de las áreas de inundación.

El uso de la optimización multiobjetivo en el drenaje urbano no está muy difundido y/o documentado. La mayor parte del trabajo que se ha realizado en este ámbito es en redes de distribución de agua potable con optimización de un solo criterio, como se puede apreciar en Dandy et al (1996), (2001); Savic y Walters (1997); Abebe y Solomatinne,(1998). Sobre problemas de objetivos múltiples Prasad y Park (2004); Kapelan et al, (2006) abordan someramente el problema. Estas experiencias han demostrado la capacidad de los algoritmos genéticos (AGs), agrupamiento-adaptación-cubrimiento (ACCO) y otros algoritmos de búsqueda aleatoria para resolver problemas de optimización en redes de tuberías. Sin embargo, existen pocas aplicaciones sobre los sistemas de drenaje urbano, usando las ecuaciones completas hidrodinámicas o ecuaciones de Saint Venant (Dorn y Ranjithan 2004; Barreto et al 2006). La ausencia de dichas aplicaciones se debe probablemente a la complejidad de las

ecuaciones a resolver; éstas requieren de soluciones numéricas mediante diferencias finitas y, por ende, requieren de mayor tiempo computacional.

El tiempo computacional es un problema asociado que debe ser abordado. Existen dos enfoques principales para superar este problema. Uno de ellos es la construcción de algoritmos más eficientes y el otro es aumentar la potencia de hardware. El último enfoque implica el uso de supercomputadoras, lo que no siempre es posible. Sólo unas pocas universidades e instituciones de investigación tienen el presupuesto para adquirir este tipo de computadoras. Una solución más económica es la utilización de un “Cluster” de PCs que permite el uso de redes de oficina. Es común encontrar cuatro o más PCs en las oficinas pequeñas y que generalmente son subutilizados. Hay aplicaciones exitosas de “Clusters” de PC para resolver problemas de ingeniería con alta demanda computacional, algunos ejemplos se pueden encontrar en Coello et al (2002) y más recientemente Kumar et al (2006) presentan un AGs para redes de distribución de agua potable con un objetivo.

El objetivo de este estudio es abordar los problemas mencionados mediante un estudio de caso en donde se necesita de la rehabilitación de un SDU. La metodología propuesta proporciona a los encargados de tomar decisiones un instrumento adecuado para la selección de la mejor alternativa de inversión y es capaz de plantear el problema teniendo en cuenta los múltiples criterios requeridos en la optimización del drenaje urbano. Además, los problemas del modelado de los sistemas subterráneos (tuberías) y de superficie son también incluidos. La gran demanda de potencia para el cálculo se aborda a través del uso de un “Cluster” de PCs. La selección de la plataforma, método de paralelización y el tipo de modelo, son de principal importancia para el correcto desarrollo del algoritmo genético en paralelo y se abordan en las próximas secciones.

Metodología

El drenaje urbano supone la gestión de una variedad de intereses y partes interesadas. El diseño de

un plan de rehabilitación debe tener en cuenta no sólo la cantidad de agua, sino también la calidad del agua y los usos recreativos (Ellis et al, 2004). El enfoque que debe ser utilizado en los planes de drenaje urbano debe ser sostenible, esto quiere decir que debe integrar el interés ambiental, económico y social en equilibrio. Encontrar ese equilibrio no es una tarea fácil, como resultado de los intereses contrarios.

Los SDU se pueden dividir en dos sistemas principales, el superficial (el sistema principal) y subterráneo (sistema secundario). Cada sistema tiene sus propias propiedades y modos diferentes de modelar y medir su funcionamiento. Por ejemplo, el sistema subterráneo tiene un comportamiento en una sola dimensión, mientras que el sistema superficie es principalmente en dos dimensiones. Los indicadores para medir el funcionamiento o “nivel de servicio”, son diferentes para dichos sistemas, principal y secundario. Ellis et al (2004) muestran una lista de indicadores de rendimiento, basado en el desarrollo de un enfoque de divisiones multinivel, los cuales son utilizados como base en el presente trabajo.

El método multicapas

Este trabajo introduce un nuevo método llamado “multicapas” que es capaz de manejar varios niveles de servicio al mismo tiempo en la optimización del drenaje urbano. Algunos niveles de servicio pueden estar en conflicto el uno con el otro. Con el fin de combinar y fusionar los resultados en una solución óptima (o un conjunto de soluciones aceptables) es necesario un proceso iterativo de optimización multiobjetivo. El proceso de optimización multiobjetivo desarrollado vincula varias capas de niveles de servicio en forma espacial y temporal, tal como se muestra en la Figura 1. Este enfoque permite definir una zona común donde los usuarios y los encargados de tomar las decisiones de inversión pueden negociar dentro de un espacio de común aceptación, donde todas las soluciones son igualmente óptimas. Los niveles de servicio se asumen como espacialmente distribuidos en toda el área de

drenaje. Por ejemplo, los daños causados por las inundaciones, el riesgo, ansiedad y la estética del paisaje, pueden ser distribuidos espacialmente en niveles de servicio.

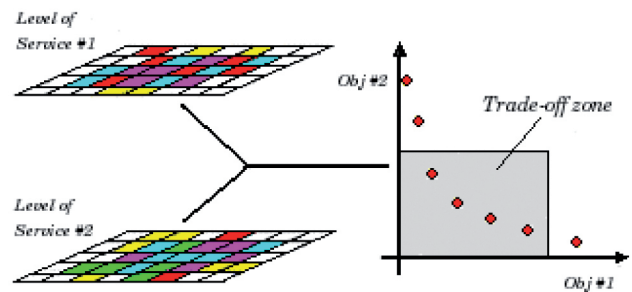


Figura 1. Optimización Multicapa para SDU.

Un optimizador multiobjetivo es usado para generar diversos escenarios que luego son evaluados mediante el uso de un modelo hidrodinámico en una o dos dimensiones. Los respectivos niveles de servicio como costos de rehabilitación o daños causados por las inundaciones o el riesgo, pueden ser calculados a partir de los resultados del modelo para cada celda y para cada tiempo y, posteriormente, integrados sobre toda el área. Un sistema de información geográfica (SIG) sirve de apoyo mediante su capacidad y funcionalidad de álgebra en mapas, para integrar los niveles de servicios a lo largo de toda la zona de influencia urbana. Este proceso debe repetirse hasta que se identifica el mejor conjunto de soluciones (Pareto). De esta manera, los encargados de adoptar decisiones y los interesados o actores podrán obtener un adecuado conjunto de soluciones que son esencialmente óptimas y permiten negociaciones dentro de una franja de seguridad y aceptación por las diferentes partes.

Paralelización del Algoritmo Genético

Uno de los principales problemas en la optimización de sistemas de drenaje urbano es el costo computacional en la solución de las ecuaciones hidrodinámicas, con el fin de estimar la profundidad del agua, la velocidad y duración de la inundación. Esto puede

ser un cuello de botella en el proceso y empeora si un modelo hidrodinámico 2D es empleado.

El principal objetivo de convertir un algoritmo multiobjetivo a paralelo, es la aceleración del tiempo computacional y la ganancia de calidad en las soluciones, al obtener una mejor convergencia al verdadero Pareto en el proceso de optimización; además de la ganancia en diversidad de soluciones. Con el fin de obtener realmente algunas ventajas con el uso de la computación en paralelo, el algoritmo serial tiene que ser dividido en un número de tareas, donde éstas deben ser independientes o “casi-independientes” en términos de datos y ejecución. Esto hace posible ejecutarlos en paralelo, utilizando otros procesadores o PCs. La Figura 2 muestra un esquema ideal, un algoritmo en paralelo a partir de un algoritmo en serie. El algoritmo es dividido en tres tareas que no son dependientes la una de la otra y cada tarea se envía a procesadores separados, ahorrando así tiempo de ejecución.

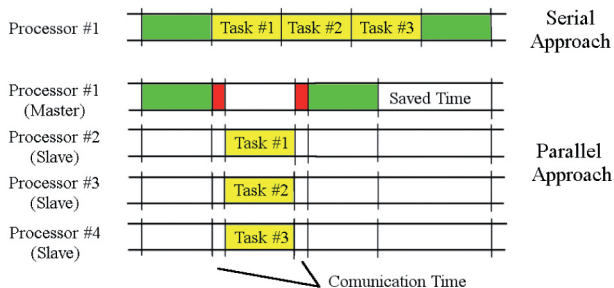


Figura 2. Diseño de algoritmo en paralelo

Granularidad

La granularidad está relacionada con el tamaño del código, el tiempo de ejecución y el tiempo usado para comunicación. Cuando el código es dividido en tamaños pequeños y su tiempo de ejecución es corto, entonces el algoritmo es definido como “de grano fino”. Si las tareas tienen un tamaño de código grande y mucho tiempo de ejecución, se les llama algoritmos de “grano grueso”.

Una cuestión fundamental que influye en el rendimiento en paralelo es el costo de comunicación. En

la mayoría de los ordenadores en paralelo, el proceso tiene que ser detenido con el fin de enviar y recibir mensajes entre los diferentes computadores. Teóricamente, a más número de tareas paralelas, mejor será el rendimiento del algoritmo paralelo; sin embargo, si se tiene en cuenta el tiempo de comunicación entre los procesadores, ésta afirmación no es totalmente cierta, porque cuando el plazo de ejecución es menor que el tiempo de comunicación, el tiempo total de ejecución del algoritmo paralelo se convierte en peor que un algoritmo en serie. Los algoritmos de “grano fino” requieren más comunicación entre procesadores, mientras que los de “grano grueso” necesitan menos. Sin embargo, el algoritmo de grano grueso tiene más código sucesivo y esto lo hace menos eficaz, incluso si el número de procesadores es aumentado.

Métodos de Paralelización

Hay tres métodos principales para convertir algoritmos evolucionarios en paralelo: amo-esclavo, isla y modelos de difusión (Coello et al 2002). Existen más clasificaciones, pero casi todos pueden encajar en estos tres modelos principales. La selección de un modelo u otro depende de la naturaleza del problema. En los modelos amo-esclavo, un procesador (maestro) se encarga de enviar los datos a los procesadores esclavos y de recoger los resultados tras su ejecución. Sólo existen comunicaciones entre el maestro y el procesador esclavo. Las operaciones genéticas se hacen en el procesador maestro y las evaluaciones de las funciones en el procesador esclavo. En el caso del modelo de Isla, la población del AG se divide en sub-poblaciones y las operaciones genéticas se ejecutan para cada sub-población en su propio procesador. En cada población evoluciona hasta converger a diferentes regiones del Pareto; por último, estas se integran para construir el Pareto definitivo. El modelo de Difusión, trata a la población en una estructura de vecindad. Las operaciones genéticas se aplican sólo entre los vecinos; según la estructura de la red, cada individuo es asignado a un procesador.

En este estudio un algoritmo evolucionario, basado en AG, fue seleccionado y transformado para

funcionar en paralelo. El algoritmo seleccionado fue el NSGA-II (Deb et al 2002). Se seleccionó por su comprobada robustez para converger al Pareto y porque ha demostrado un buen desempeño en casos anteriores (Barreto et al 2006). El algoritmo fue transformado a paralelo usando el método de “grano grueso” y fue adecuado para trabajar como amo-esclavo.

El nuevo algoritmo fue llamado NSGAXP y fue desarrollado en C, compilado utilizando GNU g++. El mismo es basado en la versión serial de Windows de NSGAX implementada por Barreto y Solomatine (2006). Como se dijo anteriormente, el cuello de botella es el modelo hidrodinámico, así que la tarea adecuada para ser ejecutada en paralelo es el modelo de evaluación de cada individuo en la población del algoritmo genético. La Figura 3 muestra el seudo-código de NSGA-II serial y cómo se transformó a paralelo.

Plataforma y selección del sistema operativo

Existen dos principales plataformas de hardware para la ejecución de algoritmos en paralelo: super-

computadoras con varios procesadores y grupos de PCs (Clusters). Las supercomputadoras son más rápidas debido a poderosos componentes en su configuración como procesadores, memoria y buses de comunicación. Sin embargo, son caras y por lo general sólo disponible para las grandes universidades e instituciones de investigación. Por otro lado, el poder de las PCs es cada vez mayor y resultan más baratas. Es común encontrar varios PCs en las oficinas de consultores, universidades, oficinas de servicios públicos, etc. El poder de cálculo total de estas PCs es usualmente subutilizado; por ejemplo, en las noches se apagan, por lo tanto, se desperdicia alrededor de 66% de la potencia instalada o son solamente usadas para navegar en la internet o escribir documentos, tareas que no saturan los procesadores actuales.

La metodología propuesta está destinada a entornos de bajo costo, como el existente en los países en desarrollo. Las oficinas tienen computadoras con diferentes sistemas operativos (OSs): Microsoft, Windows, Linux y MacOSX son los más comunes. En los países en desarrollo, el uso de código abierto es deseable pero no siempre es posible, por lo que es imperativo que la metodología pueda funcionar

Pseudo-código NSGA-II	Pseudo-código paralelizado.
Initialise population	Init evaluation procedure
Generate random pop.	
Evaluate objectives values	For parallel-task=1 to tasks
Rank using dominance	Send individual to parallel-task
Generate child pop.	Next task
Evaluate objectives values	
For generation=1 to #generations	Initialise k
With parent and child pop	For individual=1 to pop-size
Assign rank using dominance	Wait for a task result
Generate set of non-dominant fronts	Get individual
Determine crowding distance per fronts	Send individual k to slave-task
Perform selection	Increment k
Create new generation	Next individual
Perform tournament selection	
Recombination and mutation	Return from the evaluation procedure
Evaluate objectives values	
Next generation	

Figura 3. NSGAXp serial y paralelo

en múltiples plataformas. El software desarrollado, NSGAXP, puede ser compilado para ejecutarse en Linux, pero el modelo hidrodinámico, es este caso MOUSE o MIKE21, se ejecuta en el sistema operativo MS-WINDOWS.

Los PCs actuales siguen un enfoque de tratamiento secuencial, incluso los nuevos “multicore”. Con el fin de hacerlos funcionar como una máquina en paralelo, es necesaria una capa de software. Hay dos principales bibliotecas para la aplicación de procesos paralelos: “Paralelo Virtual Machine” (PVM) y “Messages Passing Interface” (MPI). PVM y MPI son muy similares y es todavía objeto de debate cuándo debería utilizarse la una o la otra. La principal diferencia es que PVM fue desarrollado pensando en ordenadores personales y una arquitectura heterogénea y MPI fue desarrollado para ser un estándar para máquinas multiprocesador. Ambos son bibliotecas de código fuente abierta y están bien documentadas. Para este estudio se seleccionó PVM, ya que la aplicación utilizando el amo-esclavo era relativamente fácil de implementar y además posee una buena gestión de errores.

Integración de los modelos

La estimación del costo de las inundaciones es función de la profundidad del agua, la velocidad y la duración de la misma. Estos son evaluados mediante el uso de modelos hidrodinámicos. Los costos de rehabilitación son una función de los diámetros de las tuberías, dimensiones de los almacenamientos y de las técnicas de rehabilitación en general. Debido a la diferente naturaleza de los programas informáticos existentes a utilizar, hidrodinámicos y de optimización, se requiere de interfaces entre ellos. En este estudio, MOUSE, del Instituto danés Hidráulico (DHI), es usado como modelo hidrodinámico base. Para comunicar el modelo hidrodinámico (MOUSE) y el optimizador (NSGAXP), fue desarrollada una interfaz, IFACE, usando Pascal. Esta interfaz tiene la tarea principal de transformar los escenarios generados por NSGAXP y elaborar el archivo de entrada de MOUSE; también calcula las funciones objetivos para cada individuo o es-

cenario. Finalmente, debe devolver los valores de las funciones objetivo al NSGAXP. La Figura 4 muestra el esquema de integración.

Estructura del Clúster

El clúster se configuró como sería normal para una red de oficinas. Esto fue hecho intencionalmente, porque no siempre es posible optimizar la red en las oficinas para trabajar en paralelo y uno de los objetivos del trabajo es llevar a cabo la optimización en una red configurada en forma normal. La configuración de una red para ser usada como clúster para cálculo en paralelo debe tomar en cuenta tanto el hardware como el software. Cuestiones como topología de red, componentes, etc., que son muy importantes para los casos de “grano fino”, están fuera del alcance de este trabajo.

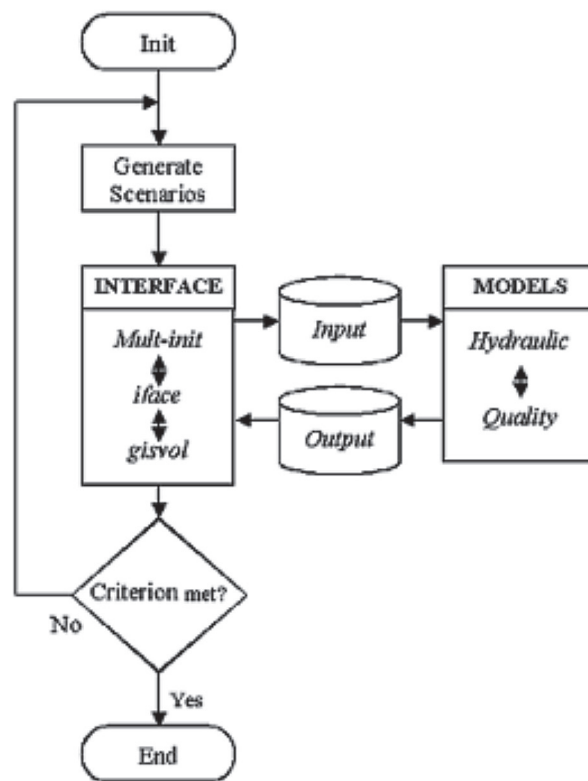


Figura 4. Componentes del modelo y su integración

Un grupo de diez computadores fue seleccionado como se puede ver en la Tabla 1. Es de remarcar

Tabla 1. Arquitectura del Cluster

Qt.	CPU	RAM KB	OS	Speed MHz
1	AMD Athlon (Laptop)	128	Debian LINUX	800
4	Intel Celeron (Desktop)	128	Win-2000	1800
4	Intel Celeron (Desktop)	128	Win-2000	2000
1	Intel P4 (Desktop)	256	Win-2000	1800
1	Intel P4 (Desktop)	256	Win-2000	2400

la heterogeneidad en las marcas, la velocidad y el sistema operativo. Esto hace más difícil de estimar el funcionamiento, pero representa realmente una configuración de red común de oficinas. Todas las tarjetas eran ethernet 10/100 MBps conectadas con tres switches HP 212M en 10/100 MBps.

Estudio de caso

Un estudio de caso fue empleado para comprobar las bondades de la metodología planteada. El mismo es un caso simplificado compuesto por doce tuberías, trece nodos y once sub-cuencas de captación (Figura 5). Para la optimización en la rehabilitación de la red se tomaron dos objetivos: reducir los costos por daños de inundaciones y reducir el costo de inversión en el remplazo o actualización de tuberías. Se parte del supuesto de que no todo el sistema se puede actualizar a la vez, debido a los elevados costos de inversión. El encargado de la toma de decisión tiene que seleccionar una solución “óptima” a partir de un determinado conjunto de escenarios (Pareto), donde se desea la reducción de las inundaciones, incurriendo en menos inversión monetaria.

Otras técnicas de rehabilitación pueden ser incluidas como tanques de almacenamientos, derivaciones, desbordamientos, etc. La ecuación 1, muestra el primer objetivo para los costos de renovación de las tuberías y la ecuación 2, el segundo objetivo para gastos por daños producto de las inundaciones en los nodos. Estas funciones de costo son normalizadas usando la combinación de diámetros de tuberías más costosa, tomadas de un catálogo

comercial, y los gastos de daño actuales por inundación que son los daños máximos esperados.

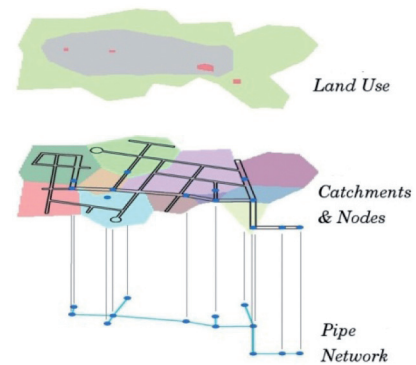


Figura 5. Estudio de caso, capas a tomar en cuenta

La modelación hidráulica se llevó a cabo utilizando el software MOUSE (1D). Esto introduce una limitación en la estimación del costo por inundación superficial, la cual es esencialmente 2D. A fin de superar este problema, las calles son modeladas como canales sobre el terreno, los cuales están conectados a través de vertederos con la red de tuberías subterráneas. Esto da una ruta de acceso preferencial superficial que emula la capacidad de transporte de las calles. Si el nivel del agua supera el nivel de la acera, el agua se distribuye en el terreno; dependiendo del volumen derramado y de una curva de volumen-área-daño para cada sub-cuenca, se determina el costo por inundación. Se trata de una simplificación del comportamiento real en 2D en la superficie.

$$f_i = \text{Min} \left(\sum_{i=1}^n (L_i * C_i) \right) \quad (1)$$

$$f_2 = \sum_{i=1}^{ncells} \sum_{k=1}^{nlu} (fc_i^k \cdot C_{max}^k \cdot dep_i) \quad (2)$$

Donde n es el número de tuberías, L_i es la longitud de la tubería y C_i es el costo de catálogo de tubo (incluyendo instalación y mantenimiento). Para la ecuación 2: f_2 son los daños de inundación total; $ncells$ es el número total de celdas en el mapa; nlu es el número total de tipos de uso de la tierra; fc_i^k es el valor de factor de daño para la celda i^b con el tipo de uso de la tierra de k^{th} ; y dep_i es la profundidad del agua en la celda i^b .

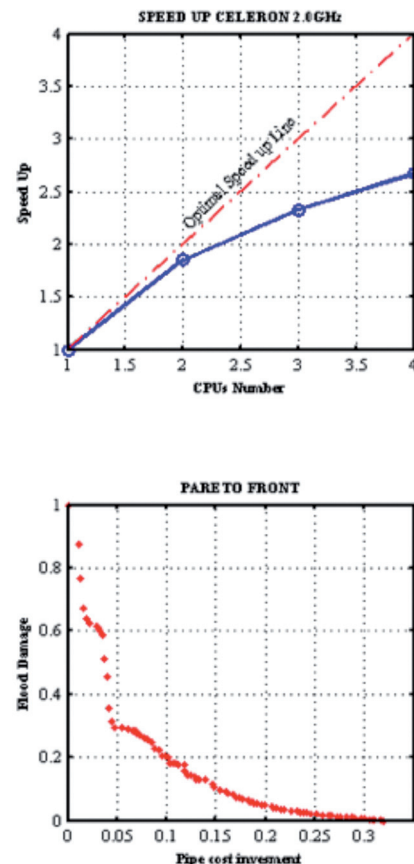
Resultados y análisis

El algoritmo genético para el caso en estudio fue configurado usando un tamaño de población de 100 individuos y fue corrido para un máximo de 50 generaciones. La probabilidad de cruce fue puesta a uno (1) y la probabilidad de mutación fue escogida según recomendaciones bibliográficas en $1/np$, donde np es el número de tubos a optimizar. Las tuberías fueron seleccionadas de un catálogo comercial que contiene un máximo de 14 costos de tuberías.

En la optimización, las funciones objetivo fueron evaluadas alrededor de 5000 veces; usando un procesador con código de optimización serial, el tiempo total de ejecución fue de 12.78 horas. La figura 6a muestra el “speed up” para 1, 2, 3, y 4 computadores del tipo Celeron de 2GHz. No muestran el resto de los procesadores adicionales porque ellos son de velocidades diferentes y esto introduciría alguna distorsión en la figura y por ende en la interpretación, el cual no estaría relacionado con la cantidad de PCs usados. Añadir más procesadores no implica un aumento lineal de la velocidad de cómputo, porque parte del tiempo de ejecución es utilizado para la comunicación entre procesadores. Más procesadores implican más tráfico en la red.

Una segunda prueba se llevó a cabo utilizando el grupo completo de 10 ordenadores. El tiempo total gastado por el grupo para las 5000 corridas fue de 4.77 horas, reduciendo el tiempo de procesamiento a

8 horas en total, lo que representa una reducción del 62,5% en tiempo de ejecución. La Figura 6b muestra el Pareto final. Los encargados de tomar decisiones pueden utilizar este Pareto como una herramienta de negociación con otros actores interesados y establecer un rango de aceptación en los que ambos puedan negociar. Asimismo, en virtud de la limitación de los presupuestos para inversión, los encargados de tomar decisiones pueden seleccionar la región de mayor impacto, donde obtengan una mejor reducción de daños por inundación por unidad monetaria invertida. Por ejemplo, invertir el 5% del costo máximo en mejorar las tuberías, reduciría en un 70% los daños causados por las inundaciones. Sin embargo, para reducir el 30% restante de los daños es necesario invertir 25% el costo máximo en rehabilitar las tuberías. Este análisis se obtiene a partir del Pareto y tendría suma importancia para la toma de decisiones.



Figuras 6a. “Speed up” para un Cluster homogéneo de 4 PCs Celeron. 6b. Pareto normalizado luego de 5000 evaluaciones.

Conclusiones

El desarrollo de un nuevo método “multicapas” en paralelo ha sido presentado y probado para ser usado en la rehabilitación del drenaje urbano. La aplicación y la integración de un conjunto de diferentes herramientas como modelos hidráulicos de optimización multiobjetivo mediante algoritmos genéticos y sistemas de información geográfica, mediante interfaces y librerías de uso público, ha sido exitosamente logrado. El nuevo enfoque ha demostrado un buen desempeño, logrando reducir el tiempo de ejecución total en 62,5%, de 12,8 horas a 4,8 horas para un estudio de caso.

El Pareto obtenido ayuda a los encargados de tomar decisiones y a los actores interesados en la selección de una solución óptima, teniendo en cuenta los diferentes objetivos presentes en la rehabilitación de los SDUs. Además, ayuda a identificar las mejores regiones de inversión que otorgan más beneficios (reducción de los costos por daños), con menos inversión.

En los casos en que el tiempo de cálculo es un cuello de botella para el análisis y tomas de decisiones, este enfoque puede aplicarse a bajo costo, evitando el uso de súper computadoras que no están disponibles para las pequeñas empresas o los países en desarrollo.

Referencias

1. Abebe A.J., Solomatine D.P. (1998). *Application of global optimization to the design of pipe networks*. Proceedings of 3rd International Conference on Hydroinformatics, pp. 989-995.
2. Barreto W., Vojinovic Z., Price, R.K., Solomatine, D.P. (2006) *Approaches to Multi-Objective Multi-Tier Optimization in Urban Drainage Planning*. Proceedings 7th International Conference on Hydroinformatics, Nice.
3. Coello, C., Van Veldhuizen, D. and Lamont, G. (2002) *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers. New York.
4. Dandy, G., Simpson, A., and Murphy, L. (1996). *An improved genetic algorithm for pipe network optimization*. Water Resources Research, 32(2), pp. 449–458.
5. Dandy, G., Engelhardt, M.O. (2001). *Optimum Rehabilitation of a Water Distribution System Considering Cost and Reliability*. Proceedings of the World Water and Environmental Resources Congress. Orlando.
6. Dorn, J, Ranjithan, S. (2004). *Generating Urban Watershed Management Alternatives Using Evolutionary Algorithms*. ASCE World Water Congress.
7. Ellis, J.B., Deutschb, J.C., Mouchelb, J.M., Schollesa, L., Revitta, M.D. (2004). *Multicriteria decision approaches to support sustainable drainage options for the treatment of highway and urban runoff*. Science of the Total Environment, 334–335, pp 251–260.
8. Kapelan, Z., Savic, D.A., Walters, G.A., Babayan, A.V. (2006). *Risk- and robustness-based solutions to a multi-objective water distribution system rehabilitation problem under uncertainty*. Wat. Sci. Tech. 53(1), pp. 61–75.
9. Kumar, S., Doby, T., Baugh, J., Brill, D., and Ranjithan, S. (2006), *Optimal Design of Redundant Water Distribution Networks Using a Cluster of Workstations*. J. Water Resour. Plng. and Mgmt. 132(5), pp 374-384.
10. Prasad T.D., Park, N. (2004). *Multiobjective Genetic Algorithms for Design of Water Distribution Networks*. Water Resour. Plng. and Mgmt. 130(1), pp. 73-82.
11. Savic, D. A., Walters, G. A. (1997). *Genetic algorithms for least cost design of water distribution networks*. J. Water Resour. Plng. and Mgmt. 123(2), pp. 67–77.