

Calendarización en Redes de Distribución de Agua

Marco Antonio Cruz Chávez¹, José Crispín Zavala-Díaz¹, Carlos Eduardo Mariano Romero², Fredy Juárez-Pérez¹, Erika Yesenia Avila Melgar¹

¹ CIICAp, Universidad Autónoma del estado de Morelos, Avenida Universidad 1001. Col. Chamilpa, C.P. 62210. Cuernavaca, Morelos, México.

² IMTA, Paseo Cuauhnahuac 8532, Col. Progreso, C.P. 62550. Jiutepec, Morelos, México.
{mcruz, erikay, juarezfredy, crispin_zavala}@uaem.mx
{cmariano}@tlaloc.imta.mx

Resumen. En este artículo se presenta una metodología para la conversión de un algoritmo genético secuencial a un algoritmo genético paralelo aplicado a problemas de calendarización en redes de distribución de agua. Se hace un estudio comparativo del problema clásico de diseño de redes de distribución de agua y su evolución en las últimas décadas. Se muestra la comparación de dos esquemas de paralelización del algoritmo: paralelización simple y paralelización funcional, mostrando las ventajas y desventajas de cada esquema. La implementación de este algoritmo paralelo permite reducir el tiempo necesario para obtener soluciones aproximadas para problemas NP Completos como lo es el problema de redes de distribución de agua.

Palabras clave: Paralelización de un Algoritmo, Paralelización Simple, Paralelización Funcional, Algoritmo Genético, Redes de Distribución de Agua.

1. Introducción

Los algoritmos genéticos son algoritmos computacionales que se basan en la evolución natural de los seres vivos. A pesar de que no garantizan encontrar la solución óptima de un problema se han aplicado con éxito, en diferentes áreas de investigación, para la solución de una gran variedad de problemas. Existen diversos estudios experimentales que muestran la eficiencia de los algoritmos genéticos para encontrar buenas soluciones en tiempos de cómputo razonables. Los algoritmos genéticos pueden llevarse a ambientes paralelos de forma relativamente fácil.

La programación paralela es un mecanismo que permite reducir el tiempo de cómputo de los algoritmos. La paralelización de un algoritmo se realiza principalmente para resolver problemas complejos [1] en los que los recursos de una sola máquina no son suficientes, o bien si pudieran resolver el problema tardarían años en encontrar una solución. La finalidad de paralelizar un algoritmo es disminuir el tiempo de

procesamiento mediante la distribución de tareas entre los procesadores disponibles. La paralelización se basa en el concepto de cooperación por parte de los procesadores, lo cual permite mejorar la calidad de las soluciones de un problema. Actualmente la paralelización de algoritmos es un recurso que puede usarse cuando se trata de resolver problemas de optimización en los que el espacio de soluciones es muy grande: problemas de tipo NP-Completo, como lo es el problema de *Redes de Distribución de Agua*.

El problema de redes de distribución de agua es un problema de gran interés para los investigadores por su amplia aplicación práctica. Durante más de tres décadas ha sido estudiado ampliamente. Se han propuesto un gran número de formulaciones y métodos de solución. Sin embargo, en la práctica, sólo se han solucionado instancias pequeñas debido a la enorme complejidad del problema. De acuerdo a la teoría de la complejidad, el problema se clasifica dentro de los problemas NP-Duros [2]. Es un problema que puede abordarse desde diferentes etapas: *Diseño, Operación, Rehabilitación y Mantenimiento*. Actualmente, la mayoría de los investigadores han realizado estudios enfocados en la solución del problema de diseño.

El problema de diseño consiste en elegir los componentes básicos que forman la red. Los componentes más comunes son: tuberías, bombas, válvulas y fuentes de abastecimiento. Las tuberías comerciales, que se encuentran disponibles en diferentes diámetros y materiales, permiten llevar el agua desde las fuentes hasta los usuarios de la red. Las válvulas reguladoras ayudan a regular la presión dentro de una red de agua. Las bombas de potencia son elementos indispensables, en caso de que la técnica de distribución sea bombeo. Finalmente las fuentes de abastecimiento son imprescindibles para la distribución de agua y pueden ser ríos, arroyos, manantiales, pozos, entre otros.

La fase de diseño de una red de distribución de agua es muy importante porque de ésta depende, en gran parte, el buen funcionamiento de la misma. Se debe elegir una topología adecuada que represente la ubicación física en la que estarán localizados los componentes de la red, los usuarios, las fuentes de abastecimiento, entre otros. Es importante también considerar factores como condiciones topográficas, características de los componentes y número de usuarios. Las redes de distribución de agua pueden representarse de forma gráfica utilizando la teoría de grafos [3]. En un grafo, los nodos o vértices representan a las fuentes de abastecimiento y a los consumidores. Los arcos o aristas representan a los elementos de conexión como son: tuberías, válvulas, bombas, entre otros.

De acuerdo con Bhave [4], la topología de una red puede ser *en serie, ramificadas y malladas*. Una red en serie es aquella que no contiene mallas ni ramificaciones; es una conexión entre dos o más nodos de forma lineal. Generalmente tienen un nodo fuente, un nodo final o sumidero y uno o más nodos intermedios. Es la topología más simple que existe para las redes de distribución de agua.

Una red ramificada es similar a una estructura de árbol. Es un conjunto de redes en serie y no contiene mallas. Estas redes presentan un nodo fuente, más de un nodo final y uno o más nodos intermedios. Generalmente, las redes ramificadas se utilizan para la distribución de agua en pequeñas comunidades rurales, en zonas industriales y en zonas de riego. En la práctica, las redes ramificadas tienen como inconveniente la suspensión del servicio en diferentes puntos de la red cuando ocurren roturas o fugas en alguna tubería. Esto se debe a que, en las redes ramificadas, sólo existe un camino para llegar de un punto a otro en la red por lo que algunos usuarios quedarían sin servicio en caso de fallas en la red.

Las redes malladas son redes que contienen ciclos o mallas lo cual permite que el agua pueda llegar a un punto de la red a través de diferentes caminos. En estas redes, la interrupción del servicio ocasionado por rupturas en las tuberías ocurre con menor frecuencia ya que el agua pueda llegar a su destino utilizando diferentes trayectorias. Por esta razón una rotura en una tubería, en general, no afecta gravemente a otros puntos de la red. A pesar de que el costo de implementar las redes malladas es más elevado que el costo de implementar las redes ramificadas se justifica su uso porque presentan mayor fiabilidad [5].

La operación de una red de distribución de agua, es un factor importante también. Un diseño correcto ayuda al buen funcionamiento de la misma. Sin embargo, no garantiza que la distribución del agua sea eficiente. En la actualidad, la forma de operar de una red es muy importante para lograr que el agua llegue a los usuarios en cantidades apropiadas evitando fugas, desperdicios o desabastecimientos. Por si misma la distribución de agua en una red representa un problema de optimización. La función objetivo consiste en minimizar el costo de distribución en base a las demandas de los usuarios las cuales son variables a lo largo de cada día.

La mayor parte de la investigación realizada en el problema de redes de distribución de agua se ha enfocado en el diseño de la red. Sin embargo, la forma de operar de una red es un tema importante que merece ser estudiado para aprovechar los recursos hídricos existentes y realizar una distribución equitativa a los usuarios. En este trabajo se aborda al problema de redes de distribución de agua desde la etapa de operación.

Este artículo está organizado de la siguiente forma: La sección 1 presenta una introducción al problema de redes de distribución de agua. También se presentan algunos conceptos sobre algoritmos genéticos y paralelización los cuales son aplicados para la calendarización de operación de una red de distribución de agua. La sección 2 presenta la descripción del problema de llevar el algoritmo genético secuencial a su equivalente en paralelo para aplicarlo a la calendarización de redes de distribución de agua. La sección 3 presenta un estudio comparativo del problema clásico de redes de distribución de agua y su evolución a través del tiempo. La sección 4 presenta la metodología de conversión del algoritmo secuencial a paralelo para la calendarización de operación de la red de distribución de agua. Finalmente, la sección 5 presenta las conclusiones del trabajo.

2. Descripción del Problema

La operación en una red de distribución de agua consiste básicamente en encontrar la forma más eficiente de suministrar agua a los usuarios cumpliendo una serie de restricciones para la red y para los usuarios. Las restricciones principales son: presiones mínimas requeridas por los usuarios, velocidades en los flujos de la red, entre otras.

En la práctica, de forma ideal la operación de la red debería brindar a los usuarios un servicio constante e ininterrumpido. Sin embargo, hasta ahora, no ha sido posible implementar este esquema de operación debido principalmente a los altos costos energéticos y a la capacidad de recursos hídricos de las fuentes de abastecimiento disponibles. De forma alterna, en un estudio realizado por Cruz-Chávez et al. [6], se propone un esquema de calendarización del servicio, en el que se garantice la operación eficiente de la red y se minimicen los costos energéticos ocasionados con la distribución del agua. Se debe dar servicio a todos y a cada uno de los usuarios logrando una distribución equitativa de los recursos hídricos. En este artículo, se retoma la formulación matemática del modelo a resolver presentado en [6]. La función objetivo consiste en minimizar el tiempo de distribución óptima del agua en una red. En este trabajo se presenta un método de solución que consiste precisamente en utilizar un algoritmo genético para la calendarización del problema de Redes de Distribución de Agua.

Los algoritmos genéticos son técnicas de optimización estocástica que trabajan mediante la generación de soluciones aleatorias. La teoría de los algoritmos genéticos fue propuesta por Holland [7] y desarrollada posteriormente por Goldberg [8] y se basa en la evolución natural de los seres vivos. Así, en los algoritmos genéticos una población se compone de un conjunto de soluciones computacionales. Cada individuo se representa mediante un conjunto de valores que describen a una solución. Cada solución está codificada en cromosomas los cuales son cadenas de caracteres que representan la analogía con las cadenas de caracteres encontradas en el DNA. Los algoritmos genéticos tradicionales utilizan la representación alfabética o binaria para las cadenas y manejan los operadores clásicos de selección, cruzamiento y mutación, Figura 1.

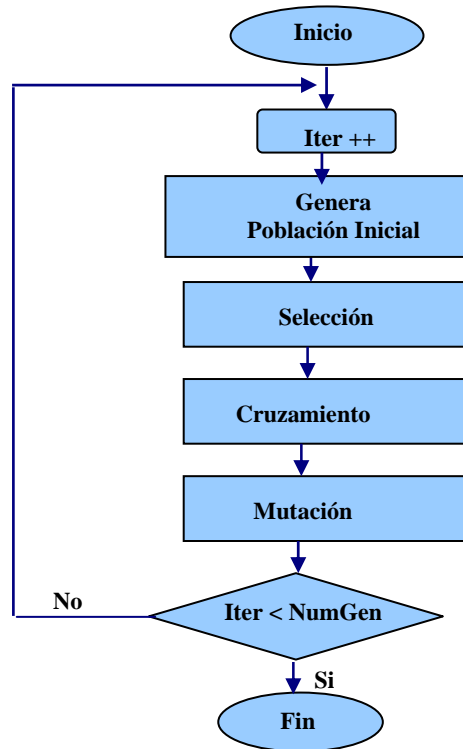


Fig. 1. Algoritmo Genético Clásico

Los algoritmos genéticos se han aplicado al problema de redes de distribución de agua. Generalmente, obtienen buenos resultados al ser aplicados en problemas complejos. Sin embargo, el tiempo de convergencia para la solución de un problema puede oscilar entre uno o varios días. Aquí surge una nueva necesidad, obtener resultados eficaces de forma eficiente. Para lograr dicho objetivo, en este trabajo se plantea una metodología para llevar un algoritmo secuencial existente a un ambiente paralelo para aplicarlo al problema de Redes de Distribución de Agua.

La complejidad del problema para convertir un algoritmo genético secuencial a un algoritmo genético paralelo es principalmente la comunicación entre procesadores. En este trabajo, el algoritmo secuencial está implementado en lenguaje C y contiene estructuras dinámicas (Figura 2) que deben enviarse, modificarse y recibirse de regreso en los nodos, en cada iteración del algoritmo. Las estructuras dinámicas permiten tener la información organizada, sin embargo utilizan apuntadores para referirse a otros datos y no se reconocen en el entorno MPI.

```

struct Fuente
{
    int NumNodo;
    int Tiempo;
    int Inicio;
    int NumOperacion;
};

struct Ciudad
{
    int NumCiudad;
    int Tiempo;
    int Inicio;
    int NumOperacion;
};

struct Calendariza
{
    Fuente **OptFuente;
    Fuente **MejorFuente;
    Ciudad **OptCiudad;
    Ciudad **MejorCiudad;
};

```

Las estructuras *Fuente* y *Ciudad* son estructuras de datos complejas porque utilizan tipos de datos básicos del lenguaje C. Los tipos de datos que utilizan (int) son primitivos.

La estructura de datos *Calendariza* es una estructura de datos dinámica ya que utiliza apuntadores para referirse a otros tipos de datos.

Fig. 2. Estructuras de Datos del problema Secuencial de Calendarización

3. Estado del Arte del Problema de Diseño de Redes de Distribución de Agua

A través del tiempo este problema ha sido estudiado por un gran número de investigadores. Alperovits y Shamir [9], proponen el problema clásico de diseño óptimo de un sistema de distribución de agua. El problema consiste en una red de tuberías las cuales llevan agua de las fuentes a los usuarios. La red puede contener bombas, válvulas y fuentes de abastecimiento. En este problema las restricciones son las demandas conocidas que deben cumplirse y las presiones en los nodos que deben respetar ciertos límites. Las variables de decisión son los diámetros de las tuberías y se basan en enfoques de variables *continuas*. Significa que entre cada par de nodos un segmento de tubería puede dividirse en dos o más segmentos con diferentes diámetros. Las variables de decisión son también, las capacidades de las bombas y la elevación de las fuentes de abastecimiento. En este enfoque clásico generalmente se utilizan redes ramificadas como caso de estudio. La formulación del problema es lineal y el método de solución propuesto es el *Gradiente*.

Después de Alperovits et al., el problema de diseño de redes de distribución de agua ha sido estudiado por varios investigadores Quindry [10], Goulter et al. [11],

Fujiwara et al. [12], Kessler et al. [13], Gupta et al. [2], Eiger et al. [14], entre otros. La mayoría de los investigadores (Tabla 1) tratan al problema como un problema *lineal* y proponen algunas técnicas de solución: modificaciones al método del *gradiente*, *Quasi-Newton*, *Branch and Bound*, entre otros.

Tabla 1. Problema de Diseño de Redes de Distribución de Agua utilizando variables Continuas

Fecha	Autores	Método	Costo
1977	Alperovits et al.	Gradiente	497525
1979	Quindry et al.	Gradiente	441522
1986	Goulter et al.	Gradiente	435015
1987	Fujiwara et al.	Quasi-Newton	415271
1989	Kessler et al.	Gradiente	417500
1990	Loganathan et al.	Heurística	412931
1993	Gupta et al.	Fletcher-Powell	407625
1994	Eiger et al.	Ramifica y Acota	402352
1995	Loganathan et al.	Heurística	403561
1997	Varma et al.	Programación Cuadrática Sucesiva	441310

La Figura 3, muestra el costo obtenido por diferentes autores para el problema de diseño de la red. A pesar de que se han empleado métodos diferentes, los resultados obtenidos en la mayoría de los casos son cercanos. Generalmente, se aplica a redes ramificadas, se utilizan componentes adicionales y la técnica de distribución del agua es mediante bombeo.

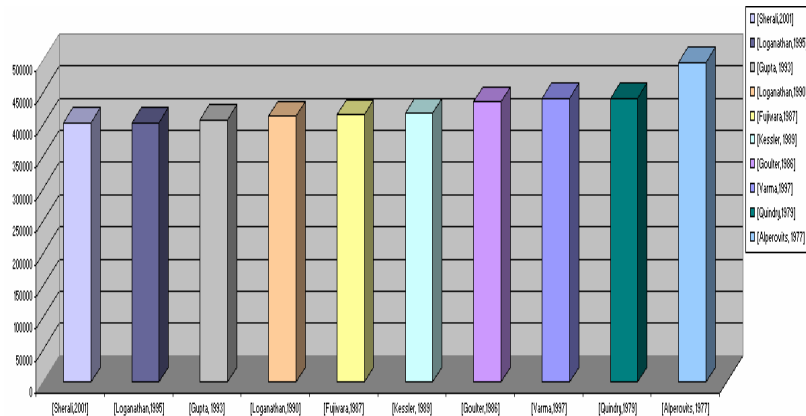


Fig. 3. Comparación de los costos utilizando Variables Continuas

Con tiempo el problema clásico se ha modificado gradualmente, de tal forma que en las últimas décadas se ha tratado como un problema *no lineal*, utilizando los diámetros de las tuberías como variables de decisión *discretas*. También, se han propuesto diversos tipos de heurísticas para resolver el problema y se han obtenido resultados experimentales similares (Tabla 2). Sin embargo, no existe la certeza de que las soluciones encontradas sean óptimas, ya que la mayoría de los métodos propuestos no tienen prueba de optimalidad.

Tabla 2. Problema de Diseño de Redes de Distribución de Agua utilizando variables Discretas

Fecha	Autores	Método	Costo
1997	Savic y Walters	Algoritmo Genético	419000
1998	Abebe et al.	Optimización Global	422000
1998	Abebe et al.	Algoritmo Genético	424000
1998	Abebe et al.	Búsqueda Aleatoria Controlada	439000
1999	Montesinos et al.	Algoritmo Genético	456000
1999	Cunha y Sousa	Recocido Simulado	419000
2001	Geem et al.	Búsqueda Armónica	419000
2003	Eusuff y Lansey	Algoritmo de la Rana que Salta	419000
2003	Matías	Algoritmo Genético	419000
2006	Reca et al.	Algoritmo Genético	419000

La figura 4, muestra los costos obtenidos con el enfoque de programación no lineal. Las variables de decisión son discretas. Generalmente la topología de las redes es mallada, la mayoría de los métodos de solución se basan en heurísticas. En general, la estrategia de distribución del agua es a través de gravedad.

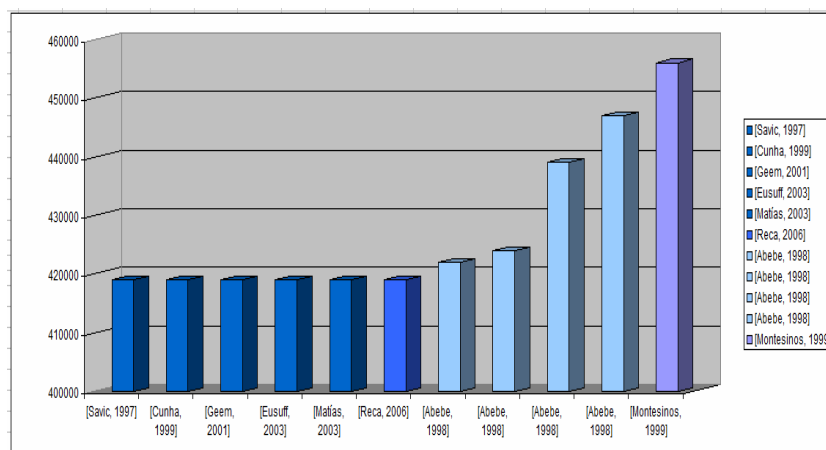


Fig. 4. Comparación de los costos utilizando Variables Discretas

En las últimas décadas, los trabajos realizados por los investigadores presentan algunas diferencias importantes. Algunos investigadores utilizan componentes en la red [6, 7, 8, 9, 10, 11] mientras que otros suponen la existencia de un modelo ideal de red [12]. Adicionalmente, algunos autores presentan topologías ramificadas [6, 7, 8, 9, 10, 11] mientras que otros consideran el diseño de redes malladas [15]. Finalmente, en algunos trabajos se utiliza la técnica de gravedad porque se considera que las fuentes de abastecimiento están a mayor altura que los nodos de demanda [4]; en otros trabajos se presenta la técnica de bombeo considerando que algunos nodos se encuentran a mayor altura que las fuentes de abastecimiento [6, 7, 8, 9, 10, 11], o simplemente porque se requiere una mayor presión en los nodos.

4. Metodología para la Paralelización Simple de un Algoritmo Genético

La paralelización de un algoritmo requiere del uso de un modelo de operación. El modelo de islas [26] divide la población original en un conjunto de sub-poblaciones distribuidas entre diferentes procesadores. Cada procesador es responsable de la gestión de la sub-población asignada de forma que ejecuta todos los pasos de la heurística, y ocasionalmente envía individuos entre islas. Aunque este modelo rompe la secuencia del algoritmo, estudios en diferentes aplicaciones han demostrado que suele mejorar al modelo maestro-esclavo, debido principalmente a que permite mantener la diversidad en la búsqueda a la vez que se mantiene un cierto grado de cooperación entre procesadores[26]. En este trabajo, una primera propuesta de paralelización para el algoritmo secuencial presentado es una metodología para la paralelización simple basada en el modelo de islas, Figura 5.

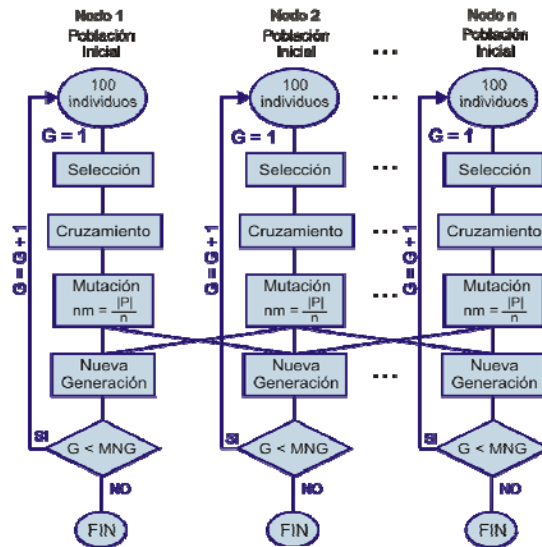


Fig. 5. Paralelización Simple

La metodología de paralelización simple, realizada en este trabajo, se basa en la idea de competencias entre procesadores y consiste en los siguientes pasos:

1. Dado un algoritmo secuencial, el primer paso de la paralelización consiste en el análisis del algoritmo para identificar las secciones críticas que consumen la mayor cantidad de recursos computacionales y que pudieran trabajar de forma independiente.
2. Una vez identificadas las funciones críticas se deben dividir de forma manual o dinámica entre el número de procesadores disponibles.
3. Posteriormente, se asigna una copia del programa a cada procesador y se le indica que funciones debe ejecutar.
4. Se define una fórmula que le permite al procesador calcular por sí mismo, el número de iteraciones a realizar en las funciones críticas que le fueron asignadas. Sobre ellas trabaja durante la ejecución del programa.
5. Se define un conjunto de datos para cada procesador y éste trabaja de forma independiente. El procesador sólo se comunica con los demás procesadores en caso de encontrar la solución óptima del problema. En este momento se termina la ejecución del programa para todos los procesadores.

5. Metodología para la Paralelización Funcional de un Algoritmo Genético

La paralelización funcional (figura 6) consiste en la división de tareas entre el número de procesadores existentes. Se utiliza el modelo maestro-esclavo el cual permite mantener la secuencia del algoritmo original [26]. Un procesador maestro centraliza la población y gestiona la selección y los reemplazos de individuos. También se encarga del envío de subconjuntos de la población a los esclavos, los cuales ejecutan tareas de evaluación, y mutación. Posteriormente, los esclavos devuelven las soluciones evaluadas al maestro.

La metodología propuesta para la paralelización funcional del algoritmo aplicado al problema de redes de distribución de agua consiste en los siguientes pasos:

1. Dado un algoritmo secuencial, el primer paso en la paralelización consiste en el análisis del algoritmo para identificar las secciones críticas que consumen la mayor cantidad de recursos computacionales. Estas secciones son precisamente las que se dividen entre el número de procesadores.
2. Se debe realizar un análisis exhaustivo del código fuente del programa secuencial para identificar el grado de cohesión y acoplamiento del programa. En caso de ser un programa con alto grado de acoplamiento, la comunicación entre los procesadores será constante para el envío y recepción de los datos que se modifican en las diferentes funciones. Se sabe que, en cómputo distribuido, las comunicaciones constantes afectan el rendimiento del programa, por lo que es recomendable en medida de lo posible evitar el exceso de comunicaciones entre procesadores.

3. Se debe analizar el tipo de datos que se enviarán a los procesadores. La existencia de estructuras dinámicas beneficia porque la información se tiene organizada y no existe necesidad de realizar varios envíos. Sin embargo debe hacerse una conversión de datos de lenguaje C a su equivalente en MPI. MPI no reconoce la existencia de estructuras dinámicas por lo que debe realizarse un proceso que equivale a serializar y de-serializar los datos. En [27], se propone una metodología para la conversión manual de un tipo de dato en C al equivalente en MPI. Al mismo tiempo se propone el uso de una herramienta para realizar la conversión de forma automática.
4. Una vez definidas las tareas y datos que se asignarán al programa, se debe identificar el número de procesadores disponibles. En base a ellos, se puede realizar una distribución dinámica tanto de los datos como del número de operaciones a realizar. Esto se hace con la finalidad de que exista un balance de carga y los procesadores terminen sus actividades casi al mismo tiempo. Con esto se logra aprovechar al máximo los recursos disponibles y se evita tiempo ocioso en los procesadores mientras se les asignan nuevas tareas.
5. En el caso de algoritmos genéticos es importante definir un proceso de sincronización. Esto se debe a que existe un número de iteraciones que deben realizarse. En cada iteración, el procesador maestro, debe recopilar los datos y actualizar la población de tal forma que se envíen datos nuevos, a los procesos esclavos. Así se mejoran las soluciones, ya que cada procesador trabaja sobre espacios de soluciones diferentes que pueden combinarse. Para la sincronización existen las técnicas conocidas como semáforos, barreras, o comunicación bloqueante definida en la biblioteca de paralelización MPI.
6. Finalmente, se deben utilizar instrucciones importantes para el algoritmo paralelo. Algunas instrucciones son propias del entorno MPI, Tabla 3.

Table 3. Funciones definidas en la Biblioteca MPI

FUNCIÓN	DESCRIPCIÓN
MPI_Init	Inicializa el entorno de MPI
MPI_Finalize	Finaliza el entorno de MPI
MPI_Bcast	Envía datos a todos los procesadores
MPI_Gather	Obtiene datos de todos los procesadores
MPI_Scatter	Distribuye los datos a todos los procesadores
MPI_Comm_rank	Obtiene los identificadores de los procesadores
MPI_Comm_size	Obtiene el número de procesadores en ejecución
MPI_Send	Envía Datos en un mensaje
MPI_Recv	Recibe Datos en un mensaje
MPI_Wtime	Devuelve el tiempo en segundos
MPI_Get_processor_name	Devuelve el nombre del procesador actual

En este trabajo, se utilizó el modelo *Maestro-Esclavo* para la paralelización funcional del algoritmo genético. Se tiene un nodo central (Maestro) que se encarga de

distribuir los datos y las tareas a los nodos esclavos. El algoritmo trabaja con una única población de individuos que se divide en partes iguales entre el número de nodos disponibles. A cada nodo esclavo le corresponde una parte de la población total, sobre la cual realizará las operaciones que le corresponden. Los nodos esclavos se encargan de realizar las operaciones de mutación sobre la parte de la población que les fue asignada. Una vez que finalizan las operaciones, los nodos esclavos devuelven sus resultados obtenidos al nodo maestro. El nodo maestro almacena las soluciones obtenidas por los esclavos en una nueva población. Sobre esta nueva población, el nodo maestro realiza las operaciones correspondientes a los operadores de selección y cruzamiento y posteriormente envía a los nodos esclavos la nueva población sobre la cual deben trabajar, Figura 6.

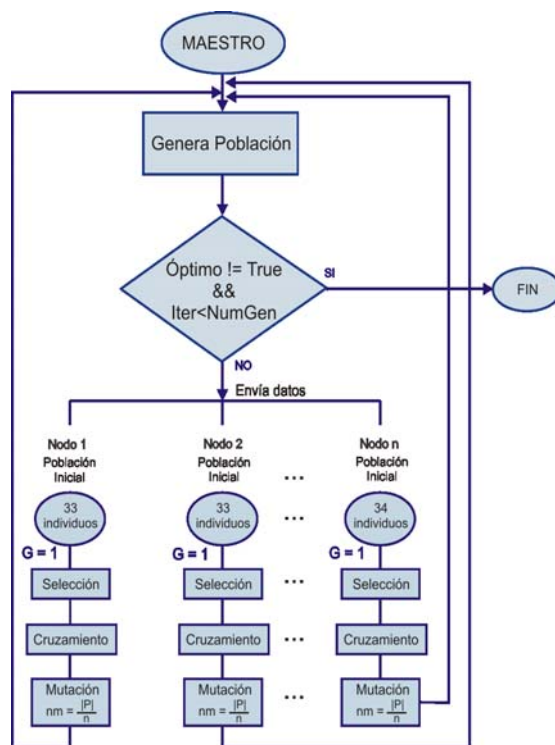


Fig. 6. Paralelización Funcional con Comunicación

La diferencia principal de los esquemas de paralelización utilizados en este trabajo, es que en el esquema funcional existe una comunicación constante entre el nodo maestro y los nodos esclavos. También se realiza la cooperación de los nodos para encontrar mejores soluciones en cada generación. Sin embargo, una desventaja de la implementación funcional es cuando algún nodo falla o simplemente se detiene, lo cual puede ocurrir con frecuencia para evitar altos costos energéticos al tener clúster de computadoras funcionando de manera in-interrumpida. Si un nodo falla o se detiene, el nodo maestro se quedará esperando los resultados de todos los nodos

esclavos y no podrá continuar con la siguiente iteración del programa. Por otra parte, la ventaja principal de la paralelización simple es que no existen dependencias entre los procesadores por lo que cada uno puede trabajar a diferente ritmo y cualquiera de ellos puede encontrar la solución óptima global. Llevar un algoritmo secuencial a un ambiente paralelo requiere de un conocimiento exhaustivo del algoritmo. De forma adicional, es necesario asegurarse que durante la paralelización los datos conserven su consistencia y que lleguen correctamente a los nodos destino durante la comunicación.

6. Conclusiones

Una de las ventajas de utilizar el cómputo paralelo es la disminución del tiempo de ejecución de un programa. El cómputo paralelo se usa como una herramienta para resolver problemas complejos de optimización combinatoria en los que los recursos de una sola computadora no son suficientes o tardarían años en solucionar un problema.

En este artículo se presentó un análisis comparativo del problema de redes de distribución de agua en las últimas décadas abordado desde el punto de vista de diseño. Se presentaron también dos metodologías de paralelización: simple y funcional. Se realizó la comparación las metodologías mostrando las dificultades, ventajas y desventajas de cada esquema.

La problemática principal de la paralelización del algoritmo genético es el envío de las estructuras dinámicas en el clúster de computadoras. Puede decirse que ambos esquemas paralelización tanto simple como paralelización funcional tienen ventajas y desventajas y dependiendo de las necesidades que se tengan se puede elegir alguna de ellas.

Las metodologías propuestas pueden ayudar en la conversión de un algoritmo secuencial a un algoritmo paralelo en un ambiente MPI. Para este trabajo se logró la conversión de un algoritmo secuencial a su equivalente en paralelo que puede aplicarse al problema de redes de distribución de agua. Un trabajo futuro que daría continuidad a esta investigación sería mostrar los resultados experimentales realizados con ambas metodologías en el problema de redes de distribución de agua.

7. Referencias

1. Blazewicz, J., Handbook on Parallel and Distributed Processing, In International Handbooks on Information Systems, Springer, 2000.
2. Gupta, I., Bassin, J.K., Gupta A., Khanna P., Optimization of Water Distribution System, Environmental Software 8, 1993, pp. 101-113.

3. Water Distribution Systems Connectivity Analysis Avi Ostfeld, J. Water Resour. Plng. and Mgmt. 131, 58 (2005), DOI:10.1061/(ASCE)0733-9496(2005)131:1(58)
4. Bhawe, P.R.(1991) Analysis of Flow in Water Distribution Networks, Ed. Technomic Publishing Company, Lancaster, Pennsylvania (EEUU).
5. R. Baños Navarro, Meta heurísticas Híbridas para Optimización Mono-objetivo y Multi-objetivo, Tesis Doctoral, Almería, España, Diciembre 2006. Disponible en: www.ace.ual.es/~rbanos/CV.html, última fecha de acceso 12 Mayo de 2009.
6. Cruz-Chávez, Empirical Transformation of Job Shop Scheduling Problem to the Hydraulic Networks Problem in a Water Distribution System, Congreso de Electrónica, Robótica y Mecánica Automotriz 2009. A publicar.
7. Holland, J. H.(1975), Adaptation in natural and artificial systems. MIT Press, Cambridge, Mass.
8. Goldberg, D.E.(1989). Genetic algorithms in search, optimization and machine learning. Addison–Wesley Publishing Co., Reading, Mass.
9. Alperovits, E., and U. Shamir (1977), Design of optimal water distribution systems, Resources Research, 13(6), 885– 900.
10. Quindry, G. E., E. D. Brill, J. C. Liebman, and A. R. Robinson (1979), Comment on “Design of optimal water distribution system” by E. Alperovits and U. Shamir, Water Resources Research, 15(6), 1651–1654.
11. Goulter, I. C., B. M. Lussier, and D. R. Morgan (1986), Implications of head loss path choice in the optimization of water distribution networks, Water Resources Research, 22(5), 819– 822.
12. Fujiwara, O., and D. B. Khang (1991), Correction to “A two-phase decomposition method for optimal design of looped water distribution networks,” Water Resources Research, 27(5), 985– 986.
13. Kessler, A., and U. Shamir (1989), Analysis of the linear programming gradient method for optimal design of water supply networks, Water Resources Research, 25(7), 1469–1480.
14. Eiger, G., U. Shamir, and A. Ben-Tal (1994),” Optimal design of water distribution networks”, Water Resour. Res., 30(9), 2637– 2646.
15. Loganathan, G. V., H. D. Sherali, and M. P. Shah (1990), A two-phase network design heuristic for minimum cost water distribution system under a reliability constraint, Eng. Optim., 15(4), 311– 336.
16. Varma, K. V. K., S. Narasimhan, and S. M. Bhallamudi (1997), Optimal design of water distribution systems using NLP method, J. Environ. Eng., 123(4), 381– 388.
17. Savic, D. A., and G. A. Walters (1997), Genetic algorithms for least-cost design of water distribution networks, J. Water Resour. Plann. Manage., 123(2), 67–76
18. Abebe, A. J., and D. P. Solomatine (1998), Application of global optimization to the design of pipe networks, in Proceedings of the International Conference on Hydroinformatics, pp. 989– 996, A. A. Balkema, Brookfield, Vt.
19. Abebe, A. J., and D. P. Solomatine (1998), Two Strategies of Adaptive Cluster Covering with Descent and Their Comparison to Other Algorithms, Journal of Global Optimization 14: 55–79, 1999.
20. Montesinos, P., A. García, and J. L. Ayuso (1999), Water distribution network optimisation using modified genetic algorithm, Water Resour. Res., 35(11), 3467– 3473.
21. Cunha, M. C., and J. Sousa (1999), Water distribution network design optimization: Simulated annealing approach, J. Water Resour. Plann. Manage., 125(4), 215– 221.
22. Geem, Z. W., J. H. Kim, and G. V. Loganathan (2001), A new heuristic optimisation algorithm: Harmony search, Simulation, 76(2), 60–68.
23. Eusuff, M. M., and K. E. Lansey (2003), Optimization of water distribution network design using the shuffled frog leaping algorithm, J. Water Resour. Plann. Manage., 129(3), 210– 225.

24. Matías, A. (2003), Diseño de redes de distribución de agua contemplando la fiabilidad mediante algoritmos genéticos, Tesis Doctoral, Universidad Politécnica de Valencia, Valencia, España.
25. J. Reca, J. Martinez, Genetic algorithms for the design of looped irrigation water distribution networks, *Water Resources Research*, Vol. 42, Almeria, España. 2006.
26. Cantu-Paz, E., *A Survey of Parallel Genetic Algorithms*, Technical Report IlliGAL 97003, University of Illinois at Urbana-Champaign, 1997.
27. Michel, M. and Devaney, J. E. 2000. A Generalized Approach for Transferring Data-Types with Arbitrary Communication Libraries. In *Proceedings of the Seventh international Conference on Parallel and Distributed Systems: Workshops (July 04 - 07, 2000)*. ICPADS. IEEE Computer Society, Washington, DC, 83.