Control Point Selection in Gas Network for Increasing Public Safety

陳勝鴻
逢甲大學資訊工程所
csh@ms.scgas.com.tw

黃秋煌
逢甲大學資訊工程所
chh@fcu.edu.tw

Abstract

Keeping safety is the primary goal for public utilities such as gas enterprise. In order to control pipeline networks area efficiently, necessary control over critical nodes is adopted by many public utilities. Using an optimal selection of such critical points, we can decide a closed area to be dominated when there is an accident happened. We found that current solutions providing control to pipeline network with a leakage of gas do not meet the optimal selection and simple needs to the public utility enterprises. On the contrary, these conventional works need a lot of related working experiences and background to complete such design goal. We derive a practical solution, an algorithm based on depth-first-search technique that is designed specifically to support the distinctive features of control point selection. With the determination of control points, we separate pipeline network into several components to dominate its flow supply efficiently and safely. The solution is designed to meet the following four high-level design goals: (1) Low Threshold to Develop a New Urban Pipeline Network System; (2) Timely Decision to Leakage of Flow in Network; (3) Low Cost of Network Maintenance; and (4) Compliance with Enterprise Security Policies. A fundamental concept of our solution is the well-calculated weighted function which corresponds to hardware cost and customer cost. Another fundamental concept is the highly developed components that could be embedded in many applications. It is easily applied to any other public utility pipeline networks. We report the results of a trial that was carried out within my own company, a gas enterprise, and discuss how well the solution met our design goals.

Key words：graph, adjacency list, depth first search, separation points, components

1. INTRODUCTION

Image the following scenario if you worked for a public utility company.

Peter Smith worked for a gas enterprise. One day, a customer made an emergency call to report a case that there is a serious leakage of gas somewhere. No matter who he was an experienced worker or not, immediate right action should be adopted to avoid any further disaster.

Feeling quickly and eagerly, Peter took out a pack of paper graphs describing about gas pipelines and customer distribution information. Peter spent many minutes to locate some possible areas dangerous. Now, a new challenge appeared, Peter must make a decision about *how to select an appropriate area to shut off the gas supply in the gas network. The selected district should not cover too huge area to decrease the customer lost due to the cut off of the gas supply. After a long period, with the help of other experienced engineer, Peter made his decision. Then he brought the necessary paper documents including the paper graph to the copier and made a copy of it. Finally, he handed over these documents to the workers who will actually maintain and fix these damaged gas pipe.*

After that, the workers got to the location where there was a leakage of gas. According to the documents information, the workers found out some valves they believed critical and shut off the valves. Also, they did some necessary maintenance to the damaged gas pipeline. After finishing these repair work, they needed to reopen these valves again. Finally, these maintenance records needed to be kept and updated the information of paper graph for future reference.

The scenario above illustrates that Peter need to face the following problems：

—He has to locate the accident's location precisely in a complicated paper graph which consists of a lot of text, lines, marks, polygons, colors, and all kinds of equipments.
—He must be very familiar with the field knowledge of pipe line network. Lacking enough knowledge will delay him to make decision.
—He needs to update paper graph information every time a maintenance record occurs.

We believe that all of job Peter do should not be so difficult. Peter should concentrate his attention to the primary job at hand; rather be distracted by grapping with the knowledge he is not familiar with.

2. WORK OF GAS ENTERPRISE

We give five features of gas work that brings about our design goal.

2.1 Work to be done using Paper document

A gas worker is a practical maintainer of gas network. Paper is a pervasive aspect of the work of gas enterprise in Taiwan. Gen [1] found that a group of 70% workers at a gas company spent about 80% of their time working with paper documents. According to our observation, paper graph is hard to understand to make a right decision to a gas accident.

2.2 Emergency Activities Are Often Unanticipated

A multi-level gas network is complex and difficult to control. Whenever there is a case reported, a gas worker must determine how badly the event will be and what kind of event to the gas pipe. Any gas worker needs to take some right actions to these events quickly and correctly without spending too much time.

2.3 Making a Right Choice is Difficulty

Since the conditions are variable, a gas worker making an appropriate decision to the gas accident will feel stressful and frustrated. A gas worker without too much working experience is hard to do the right thing to the gas accident. Even though he is an excellent gas worker, he still needs additional information of paper documents to help him to solve such critical work.

2.4 Security Levels Are Low

While working with paper graph and related documents, security level is relatively low. In practical works, we summarize three security risks：

- All levels of gas worker have the capability to access confidential gas network information without any previous permission of authority.
- An authorized gas worker took documents of gas network to the copier to make a copy. It could have a chance of leaving crucial document's copy in the public location and leads to unsecured explosion of gas equipment.
- Worker of different level should have different access level to the information of paper graph and customer's data.

2.5 Work Needs Experiences

Work of gas is the matter about safety to the public. In Taiwan, 80% gas workers have related working experience over 20 years in average. These experienced workers play an important role in making a decision to a gas accident. That means that only a fully experienced worker have the ability to handle a sudden gas event taken place.

3. AIMS OF OUR DESIGN

Taking into account the distinctive features of gas work, we derive four high-level design goals that are described in the following.

- Low Threshold to Design and Develop a new urban gas network

Designing a new developing urban area, public utilities need to preinstall public facilities including electricity power network, water supply network, and gas supply networks. In designing a new gas network, streamlined distribution of cutting nodes will help to manage the network efficiently and safely. It is also easily applied to legacy GIS [2].

- Timely Decision to Leakage of Gas (Easy Decision to Leakage of Gas)

Often, a gas event is unexpected and dangerous. We need a quick response to these events to prevent any further disaster. Providing a timely action to gas network event will increase exponent of public safety. CPS can help a gas worker to make a right decision quickly.

- Low Cost of maintenance

Control-points-selection is primary function of the solution. An optimal number of nodes selected by the solution cause a low cost in controlling the gas network. At the same time it helps to decrease the impact to customers.

- Compliance with Security Policies.

Convenient operation of network should not sacrifice enterprise security policy. Instead, It should be compliant with corporate security policies, and they should be acceptable to the information technology managers within the organization.

4. THE SOLUTION OF CONTROL POINT SELECTION (CPS)

Our solution, control point selection (CPS), was specifically designed to provide a streamlined design of newly developed gas network and provide quick response to any gas network event.

4.1 Fundamental Ideas

A gas network could be considered as a graph consisting of a lot of vertices and edges.

An edge implies a gas pipeline, while a vertex is the intersection of two or more gas pipelines. CPS solution is derived based on graph theory. CPS will find out all the separation points and separate gas network into several non-separable components. These separation vertices will be the ideal points to setup a valve controlling gas supply of area.

Depth-first-search technique is the basic foundation to design the CPS. Before we can proceed, we briefly give a detailed about DFS.

4.2 Depth First Search

The Depth-First Search technique is a method of scanning a finite un-directed graph. Since the publication of the papers of Hopcroft and Tarjan [3], it is widely recognized as a powerful technique for solving various graph problems.

- Method

The strategy followed by depth-first search is, as its name implies, to search "deeper" in the graph whenever possible [4]. Let $G\ (V,\ E)$ be a graph, in depth first search, edges are explored out of the most recently discovered vertex v that still has un-explored edges leaving it. When all of v' edges have been explored, the search "backtracks" to explore edges leaving the vertex from which v was discovered. This process continues until we have discovered all the vertices that are reachable from the original source vertex. If any undiscovered vertices remain, then one of them is selected as a new source and the search is repeated from the source. This entire process is repeated until all vertices are discovered. Making it straight, we conclude that we search as deeply as possible by visiting a node, and then recursively performing depth-first search on each adjacent node [9].

- DFS Algorithm

(1) Mark all the edges "unused". For every $v \in V$, $k(v) \leftarrow 0$. Also, let $i \leftarrow 0$ and $v \leftarrow s$.
(2) $i \leftarrow i+1$, $k(v) \leftarrow i$.
(3) If v has no unused incident edges, go to step (5).
(4) Choose an unused incident edge $v \leftrightarrow u$. Mark e "used". If $k(u) \neq 0$, go to step (3). otherwise; $(k(u)=0)$, $f(u) \leftarrow v$, $v \leftarrow u$, and go to step (2)
(5) If $k(v)=1$, stop run.
(6) $v \leftarrow f(v)$ and go to step (3).

- Concept

Consider a graph consisting of 7 nodes and 8 edges, figure 1 shows the corresponding graph. In order to represent data structure of graph, we can use three data structures, adjacency matrix, adjacency list, and indexed table to describe the graph [4]. Another data structure incidence matrix [5] is just a different way of specifying the graph. And figure 2a, 2b shows the adjacency list of graph described with figure 1. Figure3 shows the traverse order of figure 1.

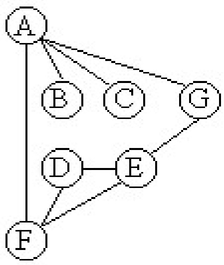

Fig.1. A graph.

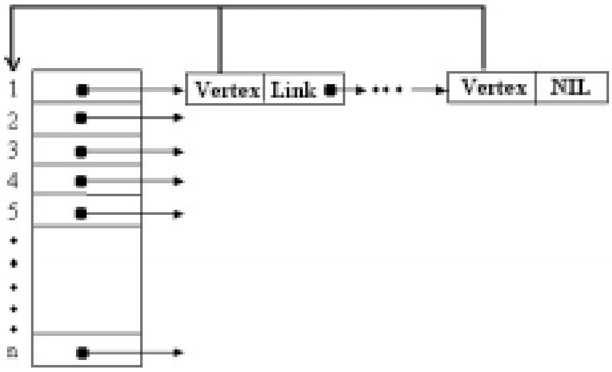

Fig.2a. Data structure of graph.

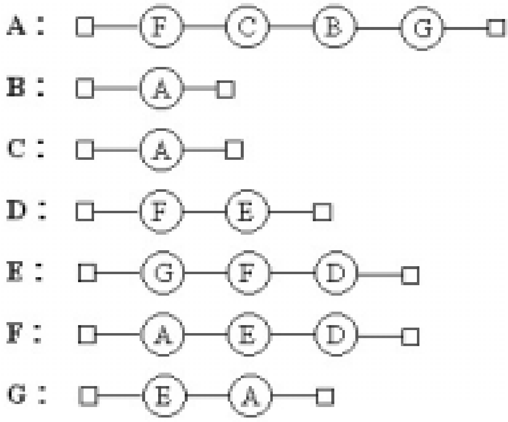

Fig.2b. Adjacency list of adjacency list from figure 1.

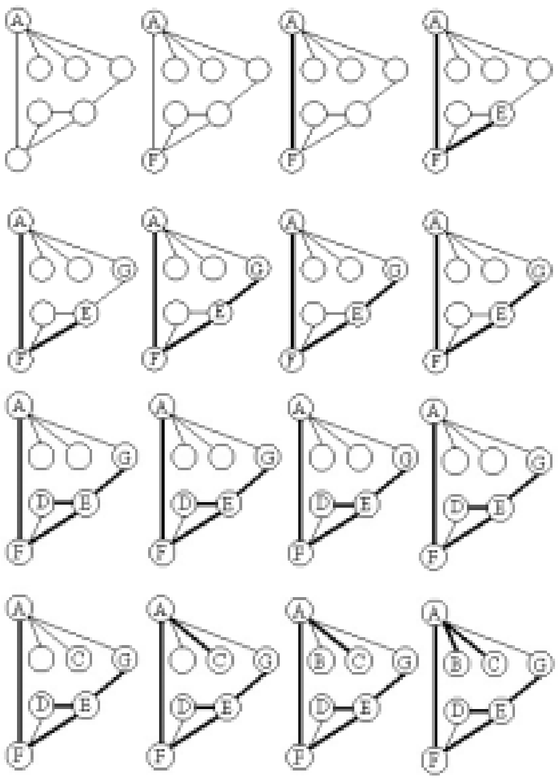

Fig.3. The search order using depth first search technique for figure 1.

- Analysis

Using adjacent list data structure, we have following property：

- The search order of DFS is based on the data structure of graph generated, even starting at the same vertex. [12]
- The number of calls to DFS is *O(V)*, since we never call DFS on a marked node, and we mark a node on entering DFS.
- The total time spent traversing adjacency lists in the while loop of DFS is *O (E)*.
- The algorithm requires *O(V+E)*. Since typically $0<N<<E<<N^2$, this is *O (E)*.

It states that every node of the list will be visited only once, so the time complexity is $O(|E|)$, namely, linear in the size of the graph.

4.3 Non-Separable Components Determination Using DFS

- Basic Concept

A connected graph G(*V, E*) is said to have a separation vertex *v* (also called an articulation point) if there exist vertices *a* and *b*, $a \neq v$ and $b \neq v$, such that all the paths connecting *a* and *b* pass through *v*. In this case we say that *v* separates *a* from *b*. A graph that has a separation vertex is called separable, and one which has none is called non-separable.

For example, in the graph shown in Figure. 4, the subsets $\{a, b\}$, $\{b, c, d\}$ and $\{d, e, f, g\}$ induce the non-separable components of the graph.

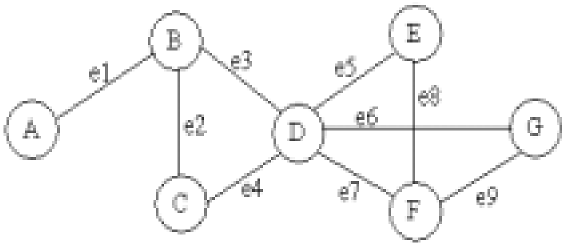

Fig. 4. Graph contains 2 separation points B and D.

- Detect Separating Vertices Using DFS

Determination of separation points is the solution to our problem. If we determined separation vertex, then there have no two non-separable components can share such vertex [6]. Some definition described below.

1. For each vertex *v*, the vertex *f(v)* from which *v* has been discovered. *f(v)* is called the *father* of *v*.
2. *Tree edges*：are edges in the depth-first graph *G*. Edge *(u, v)* is a tree edge if *v* was first discovered by exploring edge *(u, v)*.
3. *Back edges*：are those edges *(u, v)* connecting a vertex *u* to an ancestor *v* in a depth-first tree. Self-loops are considered to be back edges.
4. *k(u)* denotes the number of vertex *u*.
5. *low-point* of *v*：*L(v)*, be the least number, *k(u)* of a vertex *u* which can be reached from *v* via a, possible empty, directed path consisting of tree edges followed by at most one back edge.

■ Computing Low-point *L(v)*
At any vertex, the low point number is the minimum of [7]:

1.The DFS number of the vertex;
2.The DFS numbers of the vertex reached by a single edge;
3.The low point number of the vertex's descendants in the tree.

For example, in the graph of figure 5 with the DFS as shown in figure 4, the low-points are as following:
$L(a)$=7, $L(b)$= $L(c)$= $L(d)$=1 and $L(e)$= $L(f)$= $L(g)$=2.

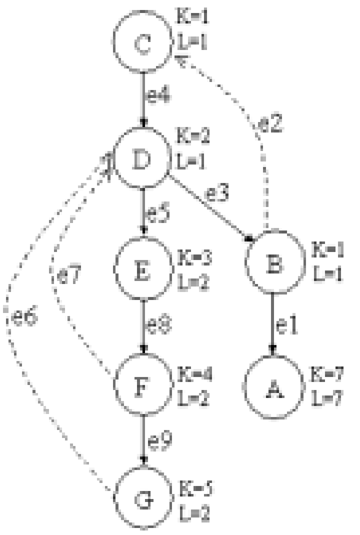

Fig. 5. Low point of figure 4.

■ Algorithm of Non-Separable Components Determination
Although the algorithm is more complicated than the simple depth-first search algorithm, but its time complexity is still O(|*E*|). This can be easily seen that each edge is still scanned exactly once in each direction and that the number of operations per edge is bounded by a constant.

Let us assume that $|V|>1$ and s is the vertex in which we start the search. The Algorithm for non-separable component determination is as follows:

(1) For every node, generate adjacency list and edge array of every link; and mark all the edges "unused".
(2) Empty the stack S.
for every $v \in V$, let $k(v) \leftarrow 0$, $l(v) \leftarrow 0$, Let $i \leftarrow 0$ and $v \leftarrow s$.
(3) $i \leftarrow i+1$, $k(v) \leftarrow i$, $L(v) \leftarrow i$ and put v on S.
(4) If v has no unused incident edges, go to step (6).
(5) Choose an unused incident edge $v \xrightarrow{e} u$. Mark e "used". If $k(u) \neq 0$,
let $L(v) \leftarrow Min\{L(v),k(u)\}$ and go to step (4). Otherwise $(k(u)=0)$
let $f(u) \leftarrow v$, $v \leftarrow u$ and go to step (3).
(6) If $k(f(v))=1$, go to step (10).
(7) If $(f(v) \neq s)$. If $L(v) < k(f(v))$, then $L(f(v)) \leftarrow Min\{L(f(v)), L(v)\}$ and go to step (9).
(8) If $L(v) \geq k(f(v))$, $f(v)$ is a separating vertex. All the vertices on S down to and including v are now removed from S; Calculate these node's weight sum;

this set, with *f(v)*, forms a non-separable component.

(9) $v \leftarrow f(v)$ and go to step (4).

(10) All vertices on *S* down to and including *v* are now removed from *S*; Calculate these node's weight sum; they form with s a non-separable component.

(11) If *s* has no unused incident edges, then output the rest of stack and stop run.

(12) Vertex *s* is a separation vertex. Let $v \leftarrow s$ and go to step (5).

4.4 Weighted Function of network

We embedded weight function into previous algorithm to reflect variation of gas network.

4.4.1 Hardware Cost

Separation points we decide in a gas network cause hardware cost that we install a valve. Since valve installation cost is the same for every separation point in a gas network, we use a constant C to present hardware host.

4.4.2 Cost of Customer Lost

A closed area we decide cause a customer lost due to a gas accident. We define a customer lost as following：

P_i : the probability of each pipe line occurs a gas accident and event.

$$\sum_{i=1}^{n} P_i = 1; \qquad (1)$$

For a gas network $G(V,E)$, every edges $u \xrightarrow{e} v$, exists a percentage that occurs gas event. The sum of all of edges percentage is equal to 1.

P_i 's value is dependent with geology, frequency of earth quake, materials used, and period of years used.

W_i : the weight of edge in a nonseparable component. Each edge weight implies customer amount in it.

L : the weight of nonseparable component, calculated by sum of every edge weights. For every nonseparable component, we have

$$L = \sum_{i=1}^{n} P_i(n) W_i(n) \qquad (2)$$

4.4.3 A Total Cost of a Component

A total cost T is defined as the cost of a non-separable component.

$$\begin{aligned}\text{Total Cost } (T) &= \textit{Hardware Cost} + \textit{Customer Cost} \\ &= m*C + L \\ &= m*C + \sum_{i=1}^{n} P_i(n) W_i(n) \qquad (3)\end{aligned}$$

4.5 Refined a Gas Network

If a node that there is no link to other node, then such node including their link has no effect to our determination of non-separable component, we can omit these leaf node and link to simplify our gas network. We derive an algorithm to solve the refinement work as following：

4.6 Brute Force search

(1) Select area considered critical.
(2) Convert a real gas network to graph.
(3) Generating linked lists of node of the graph.
(4) For every node of graph, $Links(i) \leftarrow 0$.
(5) no_node $\leftarrow$ total number of nodes in the graph
(6) $i \leftarrow i+1$, If $i \geq$ no_node, then halt.
(7) search next unvisited node u, $v \leftarrow u$
(8) $Links(v(i)) \leftarrow$ Count the number of links for every nodes.
(9) If $Link(v(i)) \leq 1$, then delete node and linked edge, go to (6)
(10) If $Link(v(i)) > 1$, go to (6).

In our experiment, another graph traversal technique, brute force search, is used to compare with our CPS. The naïve, brute force method requires *O(V(V+E))* time [7]. The brute force search algorithm we are going to use in our experiment is as following：

(1) Generate every node's linked list
(2) For every nodes of graph, let $edges(i) \leftarrow 0$ and $v \leftarrow s$
(3) For $i = 1$ to V, do

```
begin
    goto node(i);
    for link(i) ≠ null do
    begin
        edges(i) = edges(i) + 1;
        node(i) = node(i).link;
    end;
end;
```

(4) Sort array *node(i)* according to the *edges(i)*
(5) Select candidate point based on user's favorite amount
(6) Done

5. EXPERIMENTS AND DISCUSSIONS

Experiments were carried out with a PC of PentiumIII-500. The goals of the experiments were to determine whether CPS can select reasonable points in the network, and if so, how the amount of vertices and edges will affect the accuracy of selection. A secondary goal was to study the sensitivity of CPS to experimental and real-world databases. The following sections describe the data and experimental results.

5.1 Data

Experiment results are evaluated with two kinds of databases, experimental and real-world databases.

- **Experimental Database**

We form a small set of text database to described graphs. These text files are composed of different vertices and related links information that represent different gas networks.

■ **Real-world Database**

We select several areas of real gas network map and convert into text database. Each text file represents an area we will construct.

The difference between experimental and real-world database is listed in Table I.

Table I. Test Corpora

Database	Size, in files	Size, in nodes	Variety
Experimental Database	Small	Small	Heterogeneous
Real-world Database	Large	Large	Homogeneous

5.2 Results

Three sets of experiments were conducted to study the accuracy of CPS. The first set of experiments using experimental database investigated the accuracy of selection between DFS and brute force search. The first set of experiments was an initial investigation of control-points-selection with the variation settings of nodes and edges in the graph. We call these the *baseline experiments*. A second set of experiments studied the effect of CPS applied to the real-world network with different nodes number and edges number.

5.2.1 *Results of Baseline Experiments*：*DFS and brute force search on experimental database.*

The *baseline experiments* were an initial investigation of CPS. The goal of the baseline experiments was to determine whether arbitrary experimental graph data still produced accurate candidate points, and if so, how accuracy varied as a function of the total number of nodes examined. Figure 6 shows that CPS with DFS determines candidate points more precisely and economically than brute force search under the condition that certain level of nodes contained in the graph. Figure 7 shows that brute force search may have better feature of selecting candidate points in small number of edges of graph just as in figure 6. In our estimation, we find that a graph with more than 20 edges using CPS algorithm still get a reasonable amount of points to be considered. It conforms to our gas network design.

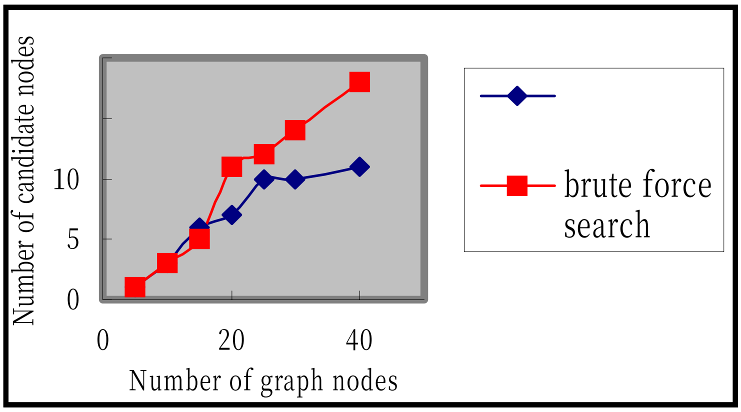

Fig. 6. Experiment of how well CPS with DFS reaches the optimal selection in different nodes graph.

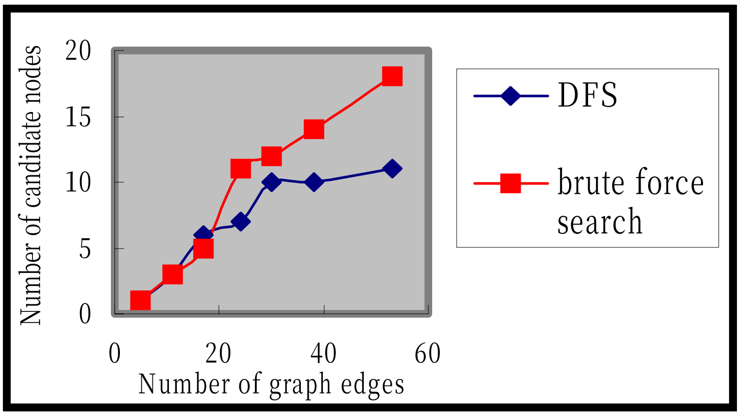

Fig. 7. DFS has better selection than brute force search in different number of edges of graph.

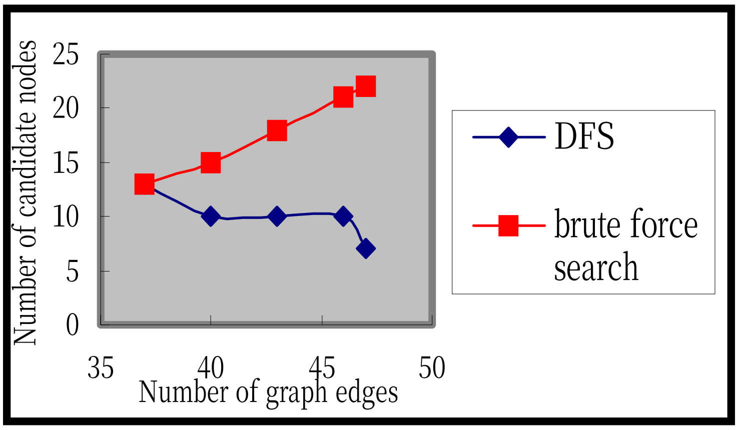

Fig. 8. Remain the number of nodes in a graph, variation of selecting points between DFS and brute force search.

As we keep the graph's nodes number a constant, Figure 8 shows that the brute force search algorithm gets more and more candidate points because of the complexity of graph, while DFS has opposite effect in selecting points.

5.2.2 *Results of DFS, Brute force search, and Manual method on real-world database.*

In this experiments, we use a part of Tai-Chung city in Taiwan as our experimental database. The district is described in figure 9. And figure 10 is the relative gas network map of figure 9.

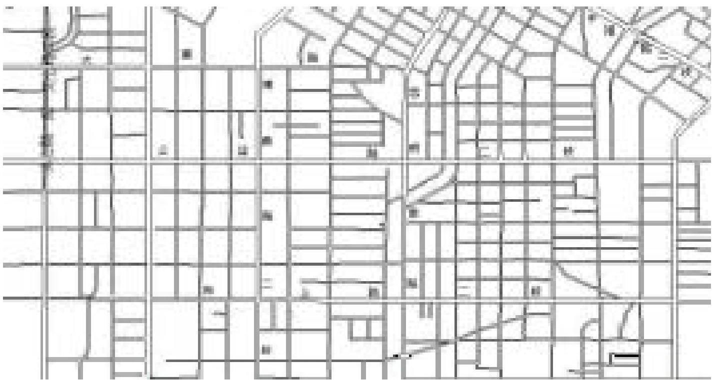

Fig. 9. The part of map in Tai-Chung City.

Fig.10. A real gas-supplied network city in Taiwan derived from the figure 9.

■ ***Result of Varying the Size of the Number of Nodes***

In the first experiment, we divide the experimental part of Tai-Chung city into five small parts. Each small part contains the measure area of 400m*1000m. We applied the first part to different approach and collect experimental data. After finishing the experiment to the first set of data, we combine first part of area with another part of area that is adjacent to the first area. The aggregation of areas forms second set of data. We feed second set of data into our experiment platform. The experiments go on in this way until we finish experiments of entire area.

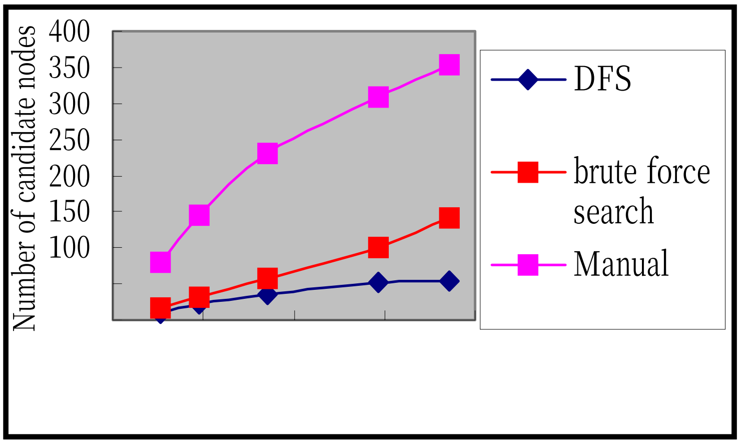

Fig.11. Measures of how well of DFS, brute force search, and manual method in a gas network with different number of nodes.

The experiment was conducted, following the same experimental methodology used in previous experiments. With a small number of nodes of gas network, DFS has a minimal growth. As we increase the amount of nodes in the gas network, brute force search was affected in a lightly way, while it has minimal effect to DFS under certain level of nodes number. It shows that manual's method leads a huge hardware cost as well as the maintenance cost in the future (Figure 11).

■ ***Result of Varying the Size of the Number of Nodes***

We suggest that DFS has better performance than any other method in varying size of edges. The experiment shows that DFS has a minimal change to the number of edges. DFS has an optimal number of nodes selected when reach some level of edges size. We believe that brute force search will have candidate points just as manual method because of the network complexity. (Figure 12).

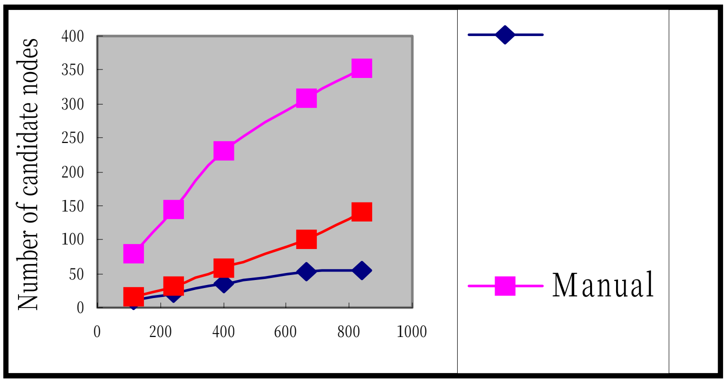

Fig.12. Measures of how well of DFS, brute force search, and manual method in a gas network with different number of edges.

5.2.3 ***Results of CPS in Experimental and Real-World Databases*.**

The last experiment verifies that DFS can satisfy our design goal, low threshold to design an arbitrary network, in both databases. Figure 13 shows that the result of real-world database is as same as the result of experimental database. Both results state that the number of candidate nodes has a linear function of nodes number except when total number of nodes reaches 250.

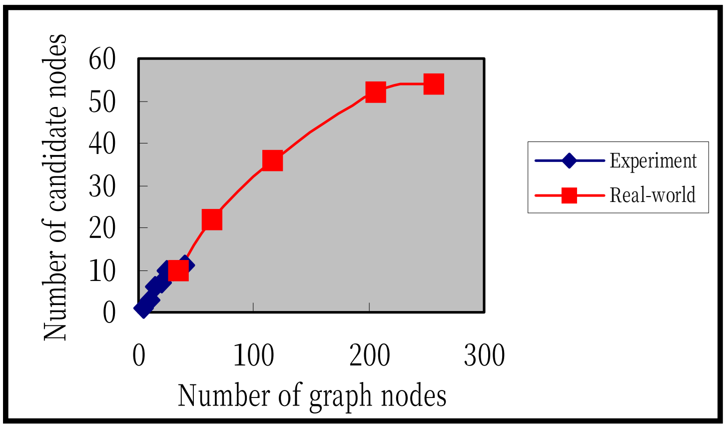

Fig.13. Measures of how well DFS can be applied in both databases.

5.3 Component Evaluation

Since the system determines separation points, we separate the network graph into several components. Based on our weight function designed, we may calculate every component's weight sum precisely. Our system will output every component weights in descending order besides the candidate nodes. According to these two output information, system operator could easily decides optimal candidate nodes with a little working experiences and related knowledge. In order to reduce the total cost of network that consists of two metric, hardware cost and customer lost cost, we merely merge two or more components into a larger component. In such a case, we can get a network with balanced weight components. At the same time, we get optimal control points through such component combination. In figure 6.9, it shows that the system generates two control points B and D under DFS method. These two control points separate the network into three non-separable components {A, B}, {B, C, D} and {D, E, F, G} respectively. According to the weight function of our design, the total weight of first component {A, B} is calculated like this, C+170*0.1=C+17. We get the total cost of other two components in the same way. The total cost of component {B, C, D} is 2C+22. The last component total cost is C+45. Here, the total weight sum of components, {A, B} and {B, C, D} is approximate to the component {D, E, F, G} total weight. Under observation of values calculated, we can merge components {A, B} and {B, C, D} into one larger component that total cost is 2C+42. Hence, we decide only one point D to be our optimal point that separates two balanced components, {A, B, C, D} and {D, E, F, G} respectively. As a result, such component combination mechanism enhances our control point selection strategy to a better performance.

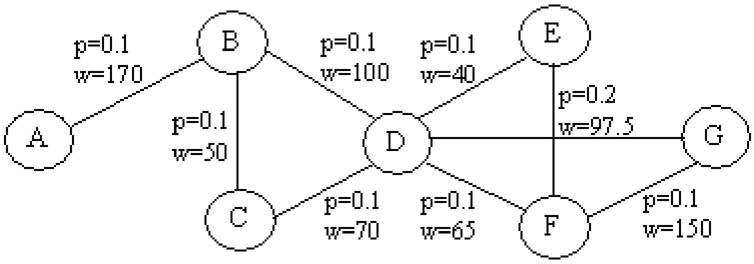

Fig. 14. This example shows the probability and customer account on each edge from figure 4 to calculate every component's weight.

5.4 Discussions

In previous experiments, we find that DFS has a better performance in selecting nodes than brute force search and manual method. But in the case of network having small number of nodes, brute force search and manual method provides better candidate nodes than DFS. Fortunately, in practical work we often need to choose our candidate nodes in a pipeline network of having more than 100 nodes or more, a measure of area 300m*400m. DFS surely reaches a better node's selection in such an area.

Keeping node number steady and increasing edge number in a pipeline network, it has little or even no impact on DFS in selecting candidate nodes. It is due to its weight function. Increasing edges makes the component more weights. Since candidate nodes are selected by means of DFS algorithm with weight function, a heavy component produces no additional candidate node.

A separation vertex separates network into non-separable components. An edge also has same effect on the network. We call an edge *e* a bridge if its deletion destructs graph *G*'s connectivity. That means that we can derive a variation of the DFS algorithm which deleting bridge, instead of deleting separation vertices. Gas enterprise in Taiwan seldom designs a network that separates pipeline. At the same time we find that most of networks applications make use of nodes to control network rather than in using edges. It might be an open and interesting question to find out practical use of bridge's determination in network applications.

6. CONCLUSIONS AND FUTURE WORKS

Our primary design goal is that a gas accident can be easily resolved by running a simple solution. Moreover, such solution also helps us to plan a newly developing gas network in a new urban. From the experiments presented in this paper we can verify that CPS can support such design goals extensively.

CPS has well-calculated weight function on it, so CPS can avoid many of geography limitations, such as measure of area, landforms, and complexity of district. CPS can be applied to older (“legacy”) GIS databases and to databases that have no GIS at all. The experimental results demonstrate that the cost of CPS, as measured by the nodes and edges required, is reasonably low, and that CPS is robust with respect to variations in parameter settings.

Finally, and perhaps most importantly, the experiments described in this paper demonstrate that CPS with DFS can be as effective, even better most of time, as manual operation of working experience. Actually, the experiments show that CPS is a very easy way to determine separation points in a gas network that will be used to install important equipments. By use of the solution, we provide a safe environment to the public in using gas energy.

Even though we have shown that CPS can get better control-points in a network, but it is still an open question on how many control- points must be sampled from these candidate points in a gas-supplied area.

The paper reported here can be extended in several directions, to provide a more feasible add-on function to any other network system like water, gasoline, and traffic control. Generally speaking, well-design network systems are such important that may affect the public’s living. We propose such solution to guarantee public’s safety.

References

[1] 簡濟化, “中國瓦斯季刊”, 中華民國公用瓦斯事業協會, vol28, 1998

[2] 施保旭, “地理資訊系統 ” ,儒林圖書公司, 1995,2001

[3] Hopcroft, J., and Tarjan, R., “Algorithm 447：Efficient Algorithm for Graph Manipulation”, Communications of ACM, No. 6, Vol 16, 1973, pp. 372-378

[4] T. Cormen, C. Leiserson, R. Rivest, “Introduction to Algorithms”, McGraw-Hill, 1990. pp. 477-485

[5] J. A. Bondy and U. S. R. Murty, “graph Theory With Applications”, Elsevier North Holland, Inc., New York, N.Y., 1976

[6] Shimon Even, “Graph Algorithm”, Computer Science Press, 1982

[7] David J. King, John. Launchbury, “Structuring depth-First Search Algorithm in Haskell”, ACM, POPL , 1995.

[8] J. Crank, H. G. Martin, D. M. Melluish, “Graphs and Networks”, Oxford University Press 1979

[9] E. Horowitz and S. Sahni, “*Fundamentals of Computer Algorithm*”, Computer Science Press, Inc., 1984

[10] Bing-Hong, Wan., “Introduction to Data Structure”, High Point information research Inc., 1996

[11] Paton, K., “An algorithm for the blocks and cutnodes of a graph.”, Comm. ACM 14, 428-275

[12] http://www.cs.mcgill.ca/~cs251/OldCourses /1997/topic26/