

Community Knowledge-based Design Patent Map System with Efficient Dissimilarity Visualization Engine: Design, Performance, and Applications

Rain Chen¹, Chao-Chun Chen², and Ting-Ting Dai³

¹Department of Creative Product Design
Southern Taiwan University, Tainan, Taiwan
ocean@mail.stut.edu.tw (corresponding author)

²Institute of Manufacturing Information and Systems
National Cheng Kung University, Tainan, Taiwan
chaochun@mail.ncku.edu.tw

³Department of Information Management,
Southern Taiwan University, Tainan, Taiwan

Abstract: Patent deployment has become competition strength for companies. The intelligence property can keep the competition advantage of a company from opponents through the patent deployment which can be visualized by the patent map technique. The patent map is an important strategic tool for establishing design strategies. Our past efforts studied the visualization transformation techniques in design patent map, and the comparisons of design patents in United States and Taiwan. Of types of patents, design patents occupy a unique patent field, since design patents are not as definitive as other patent fields. Therefore, the construction of design patent map is extremely difficult. Current commercial patent map systems visualize the patents according to non-populace attributes, even some systems constrain the number of patent to generate a patent map. Considering these scenarios, such patent map systems are insufficient for providing more objective results from populace to support more powerful evidences in law courts. A key factor to support the patent map system adopting the populace opinions is a fast dissimilarity visualization engine which can translate the dissimilarity of patents from the populace opinions to a patent map. This paper presents a GA-based dissimilarity visualization engine for the above mentioned purpose. We design a set of crossover and mutation operations based on the observations could generate patent maps with better quality. A comprehensive set of experiments are conducted, and the results reveal that the GA-based dissimilarity visualization engine indeed speeds up around 50% than the traditional method. The performance properties of our proposed method are also studied. Hence, such the engine is quite suitable for impatient users on the internet platform. Finally, we also present a case study of applying our system prototype to an industry-academy cooperation project for patent analysis and evasion.

Keywords: genetic algorithm; patent map; industry; web technology; visualization;

I. Introduction

With the arrival of knowledge economy era, enterprises are forced to treat intellectual property rights as efficient strategies in enhancing their competitiveness. Enterprises commonly use patents to achieve certain objects, including supporting decision-making, planning the use of new technology in future research and development, overcoming difficulties in some technological steps, purchasing equipment, and performing several intended purposes [9]. The World Intellectual Property Organization (WIPO) revealed that 90-95% of all the world's inventions can be found in patent documents [15, 6]. The patent literature is an enormous technical database. Many works on professional patent systems [10, 20] and patent data analytical techniques [18] are developed to analyze patent data. Some scholars further apply patent documents in different field, such patent knowledge in breast cancer medicine [2], in which authors use patent paths to establish pharmaceutical management mechanisms [16]. However, because it is difficult to interpret patent literature, professionals in different fields stop short of using the abundant patent data.

In order to cope with the great amount of patent documents, Shneiderman [17] proposed the concept of information visualization to assist directors of corporations for interpretations of patent documents. Patent map is a useful tool to exhibit the technical deployment of specific industries. Through the patent map technology, corporations can understand the state of competition or even developmental trends of the industry, so that corporations can establish their own business or development strategies [15].

Notice that the application and protection of patents are based on the territoriality principle, patent laws in different countries could be different, and thus, patents can be classified into many types. For example, in United States, The

United States Patent and Trademark Office (USPTO) offer three types of patents [19], including utility patents, design patents, and plant patents. In this work, we focus on design patents. Design patents tend to be the protection of forms and appearances, thus, comparison of design patents is more difficult than that of invention patents. The reason is that invention patents tend to have objective data for determination, such as temperature or stress; however, design patents tend to be subjective determination, such as creativity and homogeneity which is still difficult to be evaluated or differentiated by current computer techniques. Hence, human being plays the key component to evaluate or differentiate design patents. Therefore, the design patent map system is a human-machine cooperation computation. Such characteristic distinguish our work from other works that focus on invention patents [8]. Following the above mentioned property about design patents, we observe many judges on patent infringement are different from populace opinion in practice, because a judgment are decreed by only few justices. More and more patent experts believe in that collecting the populace opinion of patents as evidences will assist justices to make more fair judgments. However, no such system or platform assist barristers or patent experts to collect the statistics of populace opinion of patents so far. According to experts' experiences [7, 3, 4], such the design patent map system should equip the following two necessary properties.

- The first properties is that users can easily access and conveniently manipulate the system. The easy access is proportional to the motivation of using the system, and the number of success populace investigation would then increase. Most users has low desire to make the patent comparisons on the dedicate system and even with specified time schedule.
- The second properties is the efficiency and scalability of visualization transformation, that is, the system should respond users within acceptable response time. The too long response time would decrease users' desire on the system.

In many design courses about modern human-interface design experiences, the above two points are also important rules to increase community interaction on Internet, particularly the cloud-based service [12].

In this paper, we proposed a novel design of a community knowledge-based patent map system with an efficient genetic algorithm-based dissimilarity visualization engine, which can transform the dissimilarity among patents into a two-dimensional patent map. Hence, the two-dimensional patent map generated based on the patent dissimilarity includes the statistics of common cognition in the community, and the visual patent map is easy to interpret the patent analysis and strategy for product designers. Partial content of algorithm design is extended from our primary version [5] presented in a conference. In this work, more related topics are covered, such as complete performance study and a case study of how to apply our designed prototype. For the kernel of the patent map system, our proposed method uses a sampling comparison of patents and yet reaches high precision in a genetic-algorithm framework. Thus, the engine can generate the patent map within acceptable waiting time.

Therefore, the proposed method can be built in a platform for public access to increase the acceptance of populace. In addition, our proposed method can be scalable to large number of patents in a patent map. In our experiences, we can transform dissimilarity of hundreds of patents in a patent map, however, some commercial software packages allow only less than 100 patents, such as SPSS [11]. We also conduct a comprehensive set of performance experiments to study the transformation performance, and as well as the characteristics of the genetic algorithm to the performance. Our proposed method reduced around 50% transformation time cost than traditional methods. When more patents are transformed in a patent map, our proposed engine shows more advantage in performance. Finally, we also present a case study of applying our system prototype to an industry-academy cooperation project sponsored by Creative Service Project of Ministry of Economic Affairs, Taiwan and Winton Plastics Industries Co., Ltd¹. for patent analysis and evasion. The case study shows how our prototype assists the product designers of Winton to understand the patent advantage of their products and patent pressure of product competition.

The rest paper is organized as follows. Section II describes the system architecture. Then, we present the design of the GA-based dissimilarity visualization engine in Section III. Section IV shows the comprehensive set of performance study of our proposed method. Next, Section V demonstrates the prototype of our patent map system. In Section VI, we give a case study of applying our prototype to our industry-academy project. Finally, we conclude the paper in Section VII.

II. System Architecture

Figure 1 instantiates the user scenarios according to our condensed practical experience in design patent map systems. Assume that a number of users (populace) in the community contribute their knowledge on comparing the design patents in the web platform. Since users may not have the same available period for the patent survey, each user can proceed the comparison any time and anywhere. After comparing a pair of patents, the comparison result would be sent back to the patent database. On the other hand, the administrators can retrieve the up-to-date patent map which will be automatically generated based on the current results of populace opinions in the patent database by the system.

Figure 2 illustrates the reference architecture for transforming the dissimilarities of patents into a patent map to visualize the differences of patents. Notice that the dissimilarities of patents can be transformed to multi-dimensional space. However, in order to be able to clearly identify the differences of patents in the map, most researchers prefer two-dimensional space.

For the convenience of public access, the patent map system is recommended to design by using web technologies. While users access the system, a set of web interface components associated with the functions of the patent map systems interact with the users. Two most critical components are provided in a patent map system. The first component, called

¹More details about Winton Plastics Industries Co. can refer to its web site: <http://www.wintontile.com.tw/index.html>

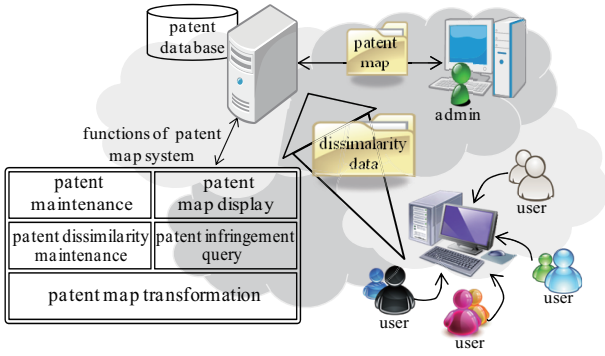


Figure 1: An illustrative Scenario of populace-used patent map systems.

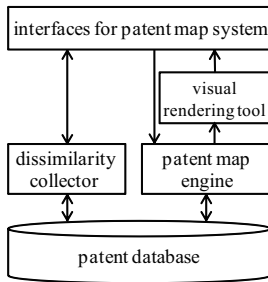


Figure 2: Reference architecture for populace-used patent map systems.

dissimilarity collector, is used to collect the dissimilarity of patents. The collected dissimilarity of patents are maintained in the *patent database* which has complete information of patents. The second component, called *patent map engine*, is used to transform the collected dissimilarity of patents into a map which visualize the relative dissimilarity of patents in a two-dimensional space. The map transformed from the dissimilarity of patents is then rendered by the *visual rendering tool* into the visual formatted data for displaying in web pages.

III. GA-based Dissimilarity Visualization Engine

After the dissimilarity matrix of patents are collected, the GA-based similarity visualization engine can generate a patent map which consists of points in a space and the distances of point pairs are similar to the similarity matrix as possible. Figure 3 shows the flow of our genetic algorithm design for dissimilarity visualization engine. The genetic algorithm processes as follows.

- Step 1.** The visualization engine generates the initial population according to the number of patents in the dissimilarity matrix.
- Step 2.** If the generation counter is less than or equal to the predefined maximal generation (i.e., *max_gen*), goto Step 3; else goto Step 6.
- Step 3.** The individuals are selected to generate new offsprings through the following three operations: 1-parent crossover, 2-parent crossover, and mutation.

Step 4. Sorting individuals of the old generation and the new generation according to their *mapdist*. Only the predefined size of population are conserved, others are discarded.

Step 5. The generation counter are increased by one, and then goto Step 2.

Step 6. Return the best individual as the result.

Step 7. End of the genetic algorithm.

In the following, we describe the components in the above mentioned genetic algorithm in details.

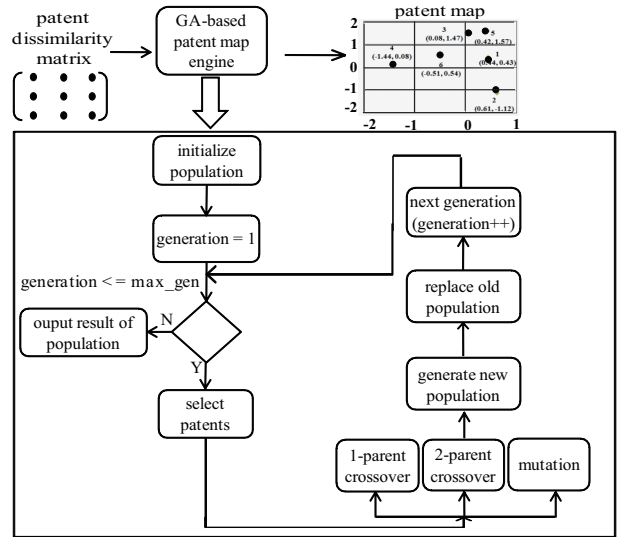


Figure 3: Genetic algorithm flow for similarity visualization engine.

Gene representation:

A patent map consists of a set of patent locations in a space. In this work, a genotype is a patent map, and a gene is a patent location. Figure 4 shows the gene representation of this work. A genotype is an array of genes, and each of which is represented as a two-dimension point.

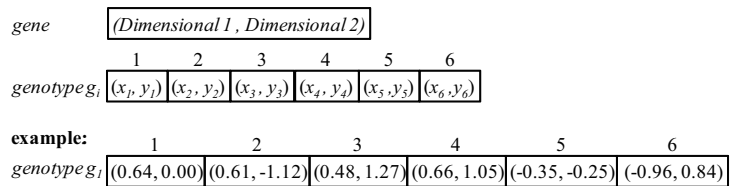


Figure 4: Illustration of gene representation.

In the traditional method, a measure to identify whether a map is similar enough to the dissimilarity matrix is through *stress*, whose basic idea is the accumulation of distance of each node pair between the map and the dissimilarity matrix. A disadvantage of using *stress* is that the computation complexity is proportional to the number of patents for comparison. Hence, we devise the *mapdist* which compares a feature set of node pairs, called *signature*, in the map comparison in order to reduce the computation cost. The signature *sign* can be express as the matrix form whose size is

the same to the dissimilarity matrix. Element (i, j) of $sign$ is set to 1 if the dissimilarity of patents i and j needs to be compared in the $mapdist$; otherwise, element (i, j) is set to 0. We define the $mapdist$ as follows.

$$mapdist = \sqrt{\frac{\sum_{i=1}^N \sum_{j=i+1}^N (D_{ij} - dist'_{ij}) \times sign(i, j)}{\sum_{i=1}^N \sum_{j=i+1}^N (dist'_{ij}) \times sign(i, j)}}$$

where the square root is just a convenience that gives greater spread to small values. Given a dissimilarity matrix D , the GA-based similarity visualization engine minimizes $mapdist$ over all point configurations (p_1, \dots, p_N) in the 2-dimension space. $dist'_{ij}$ is the distance function of points p_i and p_j , and can be obtained as

$$dist'_{ij} = \sum_{i=1}^N \sum_{j=i+1}^N dist_{ij} \times sf_m$$

$$\text{where } sf_m = \frac{\sum_{i=1}^N \sum_{j=i+1}^N D_{ij}}{\sum_{i=1}^N \sum_{j=i+1}^N dist_{ij}}$$

sf_m is a scaling factor function which can be computed once and applied to $dist'_{ij}$ computation of all patent pairs. In other words, sf_m would not incur high computation cost. Notice that calculating the $mapdist$ needs to scan all genes of the signature, the administrator may constrain the number of population in run time to further reduce the computation cost. A solution to alleviate the above challenge is to develop the engine by using the cloud computing techniques. Such part of context is beyond the scope of this paper, and thus, we leave it as the future work.

Figure 5 illustrates the difference of $stress$ and $mapdist$ from the aspect of the computation complexity. In the traditional measure $stress$, all patent pairs are considered. Assume N patents are compared in the dissimilarity matrix. Thus, the computational complexity of calculate $stress$ (referring to Figure 5(a)) is $\mathcal{O}(\frac{N(N-1)}{2}) = \mathcal{O}(N^2)$ [1]. On the contrast, only the selected patents are compared in computing $mapdist$. Assume $|sign|$ denotes the number of the selected patent pair. Then, the computational complexity of calculate $mapdist$ (referring to Figure 5(b)) is $\mathcal{O}(|sign|)$. In most cases, $mapdist$ needs less computation cost than $stress$. $mapdist$ would degrade to $stress$ if all patent pairs are selected in the signature.

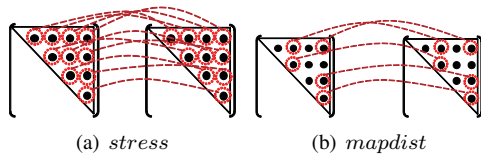


Figure. 5: Illustrating basic idea of computing $stress$ and $mapdist$.

1-parent crossover:

The 1-parent crossover operation is used to generate offsprings by imitating the similarity matrix. Algorithm 1 shows the detailed steps for 1-parent crossover. Assume N is the total number of patents. For each patent pair (i, j) and the associated element in $signature$ is marked, the positions of patents i and j will be adjusted so that the new positions will

be make the patent map structure be more similar to the dissimilarity matrix. Let p'_i and p'_j are the adjusted positions in the new offspring. Then p'_i and p'_j satisfy

$$dist(p'_i, p'_j) = d_{min} + sf_c \times dist(a_{i,j}, a_{min})$$

where sf_c means the scalability factor and can be obtained as

$$sf_c = \frac{d_{max} - d_{min}}{a_{max} - a_{min}}$$

In the above formula, a_{max} and a_{min} are the maximum and the minimum elements in the dissimilarity matrix, respectively, and d_{max} and d_{min} can be obtained as follows.

$$d_{min} = MIN\{dist(p_i, p_j) | i, j \in \{1, 2, \dots, N\}, \text{ and } i \neq j.\}$$

$$d_{max} = MAX\{dist(p_i, p_j) | i, j \in \{1, 2, \dots, N\}, \text{ and } i \neq j.\}$$

Algorithm 1: 1-parent Crossover

Input: a genotype, g_1 .

Output: new genotype g_{new} .

```
// Initialization
1 let  $N$  be the number of total patents;
2  $g_{new} = g_1$ ;
/* The dissimilarity matrix gives the
rule of the spatial-relationship of
patents to  $g_1$ . */
3 foreach patent pair  $(i, j)$  and  $signature(i, j) == 1$  do
4   adjust positions of patents  $i$  and  $j$  in  $g_{new}$ , such that
    $dist(p'_i, p'_j) = d_{min} + sf_c \times dist(a_{i,j}, a_{min})$ ;
5 update the  $mapdist$  of  $g_{new}$ ;
6 return  $g_{new}$ ;
```

Figure 6 illustrates the 1-parent crossover operation. Assume six patents in the comparison set, and the signature includes one minimum element, one random element, and one maximum element, that is, $((1, 2), (5, 6), (3, 4))$. Genotype g_1 consists six patents and associated positions $P_1 \dots P_6$. For the first patent pair $(1, 2)$, their positions P_1 and P_2 are adjusted to satisfy

$$dist(P'_1, P'_2) = d_{min} + sf_c \times dist(a_{1,2}, a_{min})$$

$$= dist(P_1, P_2) + sf_c \times dist(a_{1,2}, a_{1,2}).$$

Similarly, the same process is applied to other two pairs $((5, 6), (3, 4))$ of the signature, then a new offspring g_{new} is generated by g_1 .

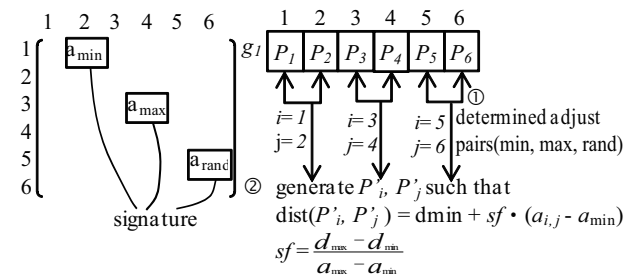


Figure. 6: Illustration of 1-parent crossover operation.

2-parent crossover:

The implementation of 2-parent crossover is depicted in Algorithm 2. The 2-parent crossover operation is used to generate offsprings by imitating other better individuals. Each time the operation is used, two individuals are randomly selected (say, g_1 and g_2) as the input of the 2-parent crossover operation. Assume $mapdist_1 \leq mapdist_2$, that is, g_1 drives g_2 . g_1 give the rule of the spatial-relationship of patents to g_2 . For each patent, say patent i in g_2 , this patent is moved to the middle of patents $NN(g_1, i)$ and $NN(g_2, i)$ in g_2 , where $NN(g_x, i)$ means the nearest patent of patent i in genotype g_x . These modifications on g_2 are used to generate a new offspring. In order to keep the diversity, one individual can be selected to proceed 2-parent crossover process from first half of the individuals sorted by the mapdist.

Algorithm 2: 2-parent Crossover

Input: two genotypes, g_1 and g_2 with mapdist $mapdist_1$ and $mapdist_2$, respectively.

Output: new genotype g_{new} .

// Initialization

1 Assume $mapdist_1 \leq mapdist_2$, that is, g_1 drives g_2 ;

/* Patent-wise crossover: in order to bring the advantage of g_1 to g_2 . */
 /* That is, g_1 give the rule of the spatial-relationship of patents to g_2 . */

2 for ($i = 1$ to N) do

 // move $g_2[i]$ to the middle of $NN(g_1[i])$ and $NN(g_2[i])$. a_1 and a_2 are index keys

3 $a_1 = NN(g_1, i)$;

4 $a_2 = NN(g_2, i)$;

5 $g_{new}[i] = \left(\frac{g_2[a_1].x + g_2[a_2].x}{2}, \frac{g_2[a_1].y + g_2[a_2].y}{2} \right)$;

6 compute_mapdist(g_{new});

7 return g_{new} ;

Figure 7 shows the basic idea of the 2-parent crossover operation. In the example, genotype g_1 has lower mapdist than g_2 , thus, the goal of the operation is to improve the quality of genotype g_2 by imitating the relationship of nodes in g_1 . The operation would scan each gene of g_1 and find out its nearest neighbor (point), and then, adjust the corresponding point in g_2 such that its nearest neighbor is the same as that in g_1 , as shown in the figure.

Mutation:

The mutation operation is designed to avoid local optimum which is frequent occurred in genetic algorithms. The Gaussian mutation is applied to our designed genotypes to generate new offsprings. The detailed steps is shown in Algorithm 3. In the beginning, a genotype is considered to proceed the mutation operation by a random filter. If the genotype is selected, the position of each patent is then adjusted by an offset of a Gaussian random number. Notice that we generate a Gaussian random number for each patent to increase the diversity of individuals in the genetic algorithm. Figure 8 shows the idea of the mutation operation. For each gene of a genotype g , the associated point is moved to a nearby location with offset generated by a pair of Gaussian numbers (i.e., for horizontal axis and vertical axis).

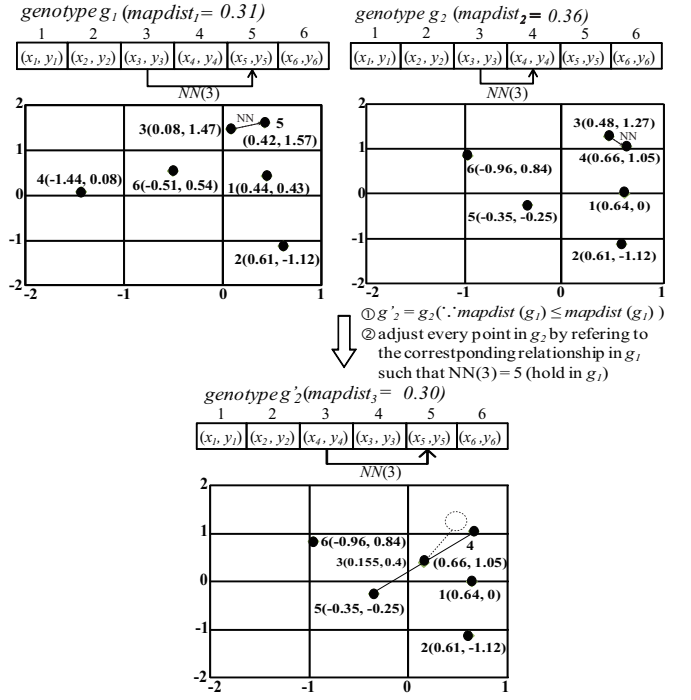


Figure 7: Illustration of 2-parent crossover operation.

Algorithm 3: mutation()

Input: genotype g .

Output: genotype g_{new} .

// Initialization

1 $g_{new} = \text{null}$;

2 $\delta = \sqrt{\min(g)}$;

3 rand_seed();

4 if ($\text{rand}() < \text{mutation_threshold}$) then

5 for ($i = 1$ to N) do

$g_{new}[i].x = g[i].x + (\delta \times \text{Gaussian_rand}());$

$g_{new}[i].y = g[i].y + (\delta \times \text{Gaussian_rand}());$

6 compute_mapdist(g_{new});

9 else

10 $g_{new} = g$;

11 return g_{new} ;

Fitness function:

Recall that one of the rules of thumb for a website design is to response to users as soon. In order to achieve the goal, we use the fitness function to control the execution time of the engine. The fitness function is controlled by a predefined parameter, called max_gen , which stands for the maximal generation in the evolution process. More specifically, once the generation count reaches max_gen , the fitness function would terminate the evolution process, and then return the current results in the genetic algorithm.

An alternative of the fitness function is designed based on $mapdist$, and the goal of such design is to generate results whose accuracy between the dissimilarity matrix and the patent map is within a predefined standard. Administrators can revise the design of the fitness function according to their needs. In this work, we only consider the max_gen -based fitness function, and the evaluation of the $mapdist$ -based fitness function is left as one of our future work.

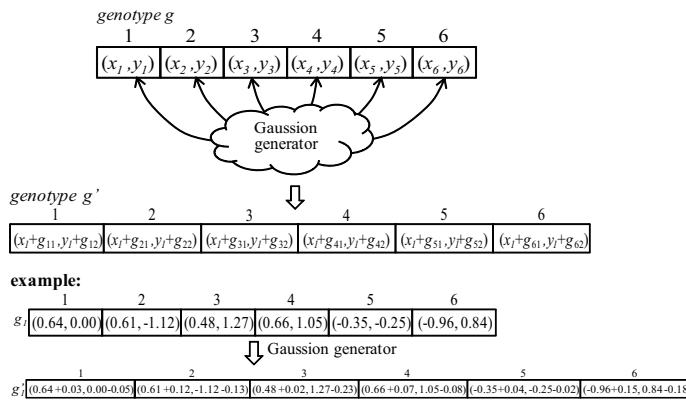


Figure 8: Illustration of mutation operation.

IV. Performance Study

In this section, we conduct a comprehensive set of experiments for studying the performance characteristics of our designed GA-based dissimilarity visualization engine. Before discussing the experiment results, we describe the simulation settings used in the simulation.

A. Simulation Settings

In our simulation, we maintained 100 real patents [8, 3] with their related description and basic information in the patent database. According to our experience for different needs of customers, the number of patent set in the experiments varies from 20 to 100. Notice that current SPSS cannot transform the dissimilarity visualization with more than 99 patents, thus, the experiments show our system has higher scalability. The default signature pattern is (9, 0, 0), which means we select the patent pairs as signature set from 9 minimum elements in the dissimilarity matrix. Later, we will show this setting does not favor our performance, thus, the results should be under fair comparisons. The prototype is built on a general personal computer which equips CPU of Intel Pentium 4 3200 MHz and 2.5GB RAM. The traditional method for transform the dissimilarity visualization is proposed by Kruscal (denoted as Kruscal method) [14, 13], and it is implemented in the simulation for comparison.

B. Experimental Results

Figure 9 shows the first experiment on studying the time consumption of the designed GA-based dissimilarity visualization engine. Our method is compared to the engine based on traditional multi-dimensional scaling which is implemented based on Kruscal's method [14]. From the figure, we can see our GA-based dissimilarity visualization engine needs only one half of computation time than that of Kruscal's method. The benefit comes from that our method can control the computation cost by setting the parameters of initial population and maximal generation. In addition, the increasing rate of our method is slower than that of Kruscal's method. Hence, our method is much more suitable for large number of patents, and the response time is also acceptable for web environment.

The second experiment is to study the influence of generations in the genetic algorithm to the mapdist for different

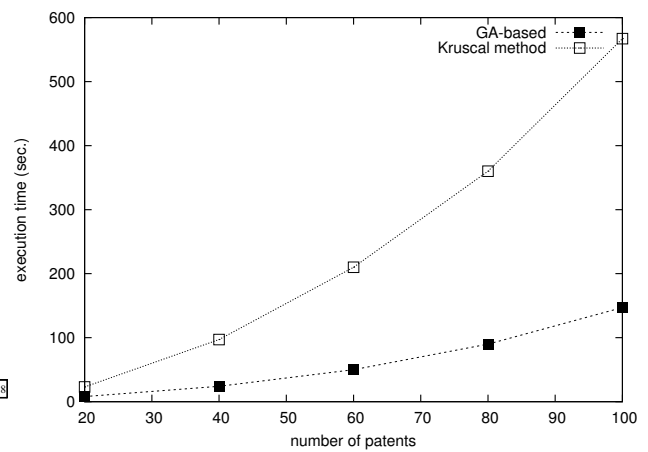


Figure 9: Comparison of time consumption between two engines.

patent sets, and the result is shown in Figure 10. For different patent sets, we can see the each curve can perform stable within ten generations. This means the designed genetic algorithm can obtain the suboptimal results rapidly. In addition, we observe the patent set of 100 patents has higher mapdist than that of 20 patent. Such result is reasonable because it is more difficult to find a patent map close to a dissimilarity matrix of large number of patents. In the result, the difference between 100 patents and 20 patents is limited, and thus, the quality of generated patent map of 100 patents is acceptable for most clients.

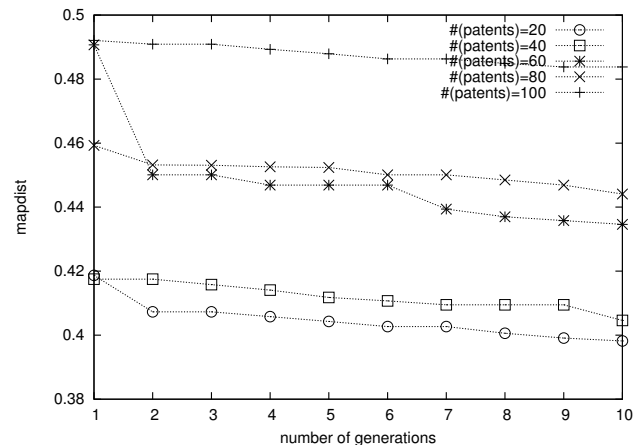


Figure 10: Impact of the generations to mapdist.

The next experiment is to study the quality of the designed genetic algorithm-based engine compared to that of traditional Kruscal's method. Figure 11 shows the comparison result. In this experiment, we use the stress value as a basis in comparison. This is because mapdist is invented and used only in our method. Thus, the patent map generated by our method is further evaluated to obtain the associated stress value. As our above mentioned on the relationship between the mapdist and the number of patent set, we can see the stress values of both methods increase as increasing number of patents. Even our method has higher stress value than the Kruscal's method, the difference is very limited for various sizes of patent set. That is, although our method only samples

some patents (signature) to compare to the dissimilarity matrix, the generated patent is yet close to that generated by the Kruscal's method. Therefore, applying the sampling mechanism, *mapdist* to the genetic algorithm obtains remarkable results.

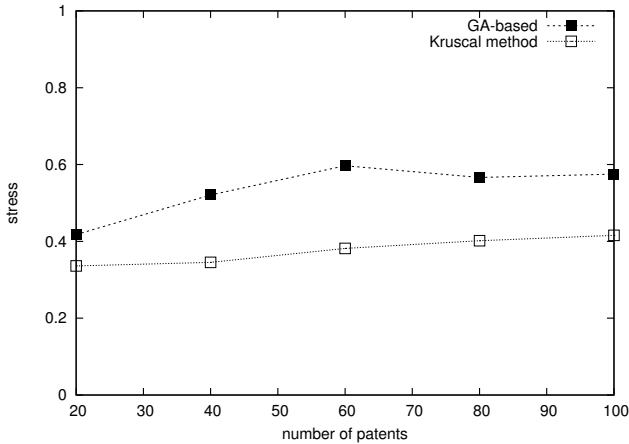


Figure 11: Comparison of stress value between two engines.

The last experiment is designed to observe the influence of the signature patterns and number of signature to the proposed method, and the result is shown in Figure 12. In order to obtaining more precise results, we again use the stress value as vertical axis, and further evaluate the stress value of the generated maps in the experimental results. The signature patterns in our experiment covers different situation as possible, thus, distribution of signature pattern includes all minimum values, all random values, all maximum values, and mixed values. From the results, we can see that most signature patterns can reach around 0.4 (while the stress value of the Kruscal's method is around 0.35), so, it is very easy for administrator to choose a signature pattern in our system for applications. In addition, we found that signature pattern $(X, 0, 0)$ has worst stress value among all signature patterns. Hence, our experiment setting $(9, 0, 0)$ is not a prejudice to our method in the experiments. In case other signature patterns are selected, the better performance results would be obtained.

V. Demonstration

In this section, we demonstrate our prototype whose design are described in previous sections. Then, we also give a real case study about applying this prototype in an industry-academy cooperation project in the next section. In order to offer convenient access and usage of a design patent map system, we develop the prototype by using web programming with a database system. The physical development platform include PHP 5.2.6 based on Apache server 2.2.8, and the patent database is built on MySQL 5.0.51a. These development tools are very popular and can cross platforms, thus, our prototype can be deployed on almost any kinds of general PC platforms which have enough computation capacity, networking module, and support the above mentioned tools. Currently, our prototype can be accessed at

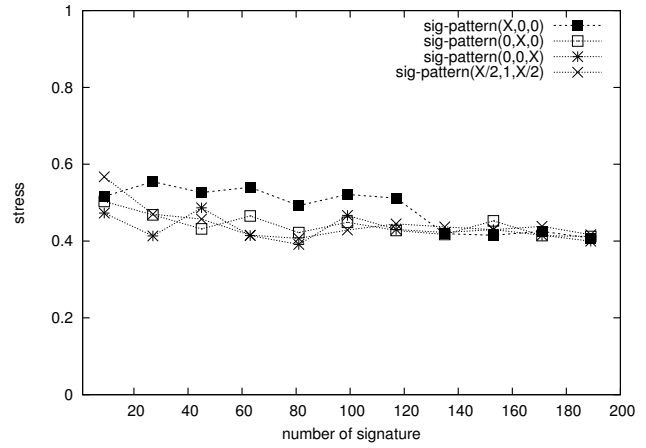


Figure 12: Impact of the different signature patterns (sig-pattern).

<http://cis.mis.stut.edu.tw/PTMS/>.

Figure 13 depicts the patent dissimilarity survey from populace. Recall that there are two types of patent pair selections in our framework, including the table view and the matrix view. In the figure, we show the comparison process by through the two types of patent pair selections. The first subfigure is the screenshot of the patent pair selections by the table view. A user can select two patents for comparisons through the assistance of thumbnail patent photos, for example, patents 1 and 6 are selected in the first subfigures. Then, the prototype will guide the user to compare the dissimilarity of selected patent pair, as shown in the second subfigure. The third subfigure shows the screenshot that a user can determine to select a patent pair by either the table view or the matrix view. the forth subfigure shows the screenshot of the matrix view.

Figure 14 shows the patent map with a visual tool. The location data in the figure are from the patent transformation engine. From the patent map, patents that most participants deem they are similar are placed closely. For example, patents 11 and 5 are very close which means most participants deem they are very similar.

VI. Case Study: Applying PTMS to Patent Analysis and Evasion

In this section, we present a real case study of applying our PTMS to an industry-academy cooperation project which is sponsored by Creative Service Project of Ministry of Economic Affairs, Taiwan and Winton Plastics Industries Co., Ltd. (Winton for short) under Grant 110527. Winton Co. is one of leading companies in producing ground mats, and designs certain products to satisfy market demands every year. Currently, Winton Co. finds some patents of related techniques are hold in competitors, and these patents would make their invented products involve in the intellectual property issues. Due to the lack of patent analysis during the design period, Winton Co. either spent costly expense for patent authorization or abandoned these new products in past years. Hence, the purpose of this cooperation project is to analyze the related products and patents, and determine some patent evasion strategies to avoid the patent authorization for their

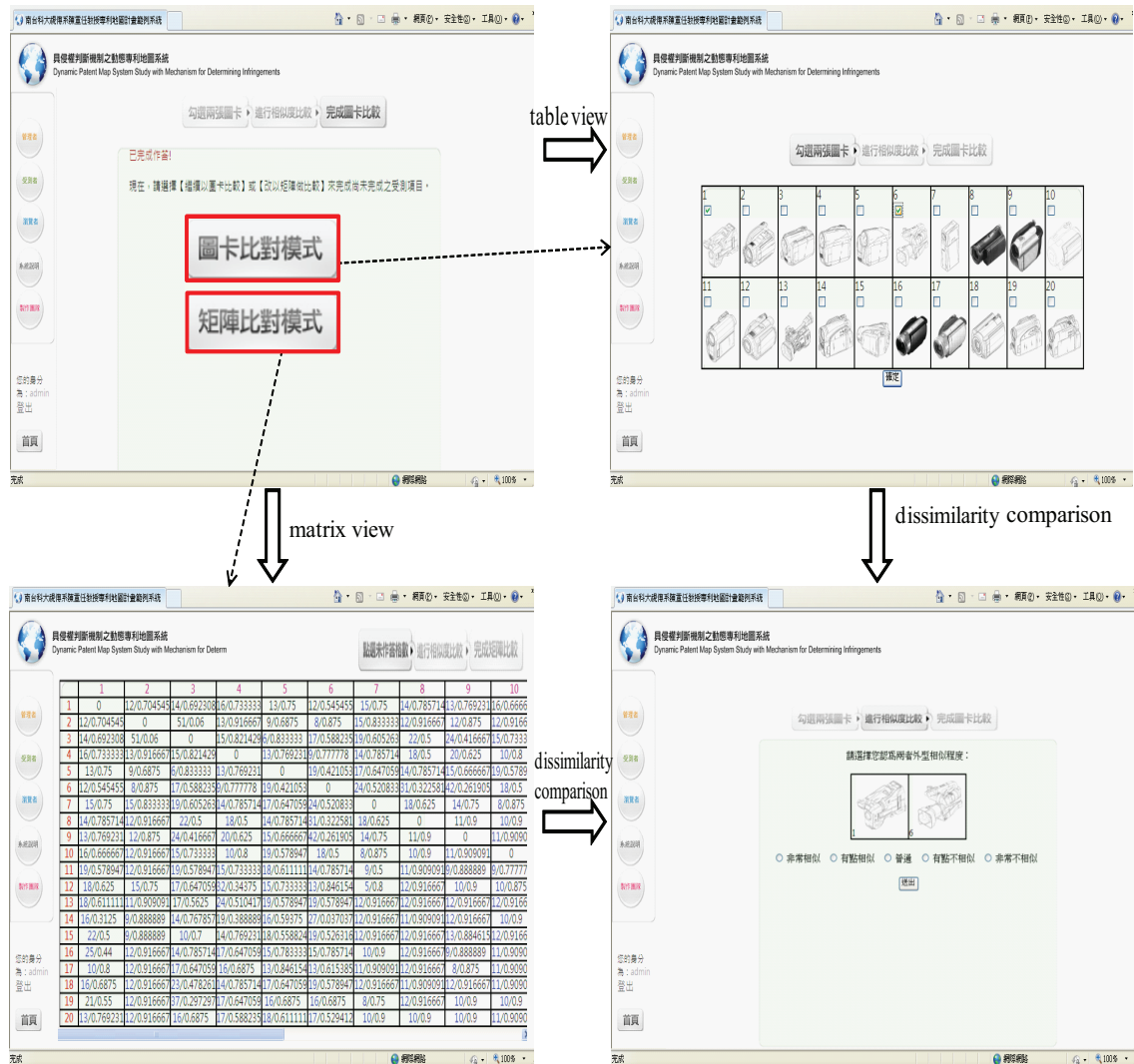


Figure 13: Screenshots of the patent dissimilarity comparison.

new products by using our proposed PTMS. Figure 15 shows three design products (namely, $D_1 \sim D_3$) out of 15 candidates for patent analysis research in the business meeting hosted by Winton Co. According to our patent search report for surveying over 2,000 patents, five US patents are possibly similar to $D_1 \sim D_3$, including US7896571B1 (X_1), US2740167 (X_2), US7856784B2 (X_3), US6516579B1 (X_4), US4426820 (X_5), where X_i is our assigned analysis code number to each patent. From our manually search report, $X_1 \sim X_4$ are similar to D_1 , X_5 is similar to D_2 , and no patent is similar to D_3 . In order to explore the relationship among the three products and five patents by using our proposed PTMS, we collect the community knowledge for each pair of products/patents, and the result is shown in Figure 16. In the figure, $D_1/D_2/D_3/X_1/X_2/X_3/X_4/X_5$ are represented as 1, 2, ..., 8, respectively. Notice that for confidential issues, we only show results of few community members to demonstrate part of our project results. The scale of our actual investigation is much greater than that in this case study. After obtaining the above dissimilarity matrix, we can then obtain the patent map of three products and five patents, as shown in Figure 17. The patent map is generated by the underlying GA-based dissimilarity visualization engine

presented in previous sections. From the figure, we found that these three products and five patents are distributed uniformly. This indicates that these products and patents are not similar to each other according to the cognitions of participant community members. Obviously, such information obtained from PTMS offers a different viewpoint from the experts in Winton Co., and thus, these results can be further studied for more details.

Our PTMS also offers additional function for display nearest patents of a specified patents, as shown in Figure 18. In the example of the figure, we query for three nearest patents of D_1 . Then, PTMS indicates patents 4, 5, and 6 are results with their distances to D_1 in a table, and uses arrows to clearly indicate their relationship in the display area of the web page. Repeating the process, we can learn the three nearest patents of D_2 and D_3 , respectively. From these results, designers in Winton Co. clearly know the novelty of their products. In addition, designers can further understand which product has most novelty and suggest the company to apply patents for this product with the highest priority. These information is also available for company to manage resources on applying patents under the limited budget constraint. Finally, PTMS can be also used to suggest experts certain products that have

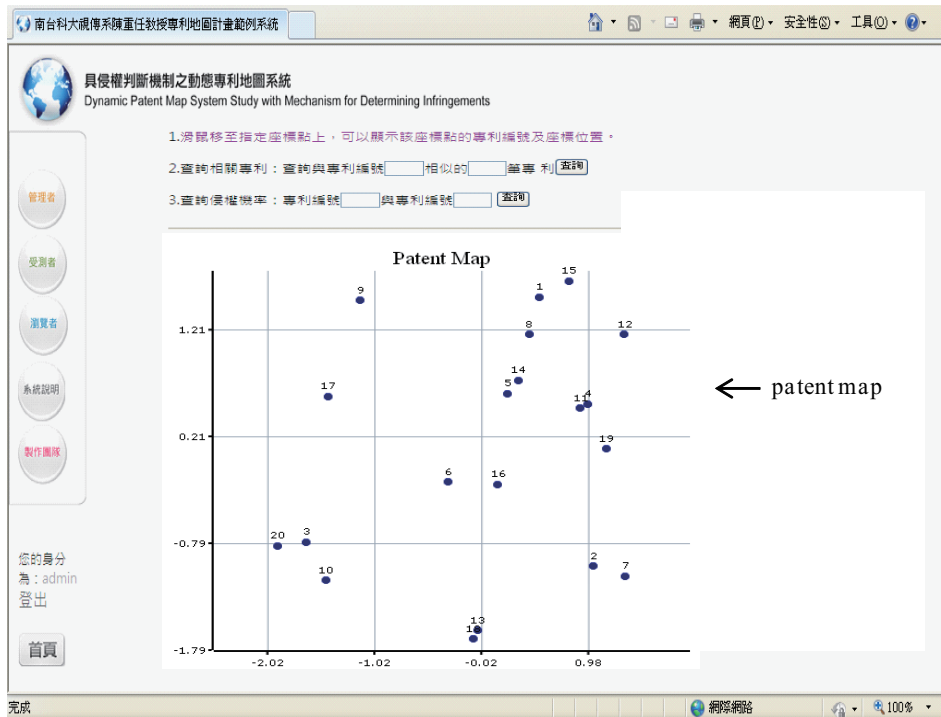


Figure 14: Screenshots of the patent map display.

least patent pressure of competition.

VII. Conclusions and Future Work

As the increases of industry competitions, patent deployment becomes the critical mean of a company to protect the products and expand the market. A patent map system can visualize the similarity of patents, so that users can easily understand the possible infringement according to the generated patent maps. In this paper, we proposed a novel community knowledge-based patent map system equipping with an efficient genetic algorithm-based dissimilarity visualization engine, which can transform the collected dissimilarity of patents into a visual map. Our proposed method is designed based on the architecture of the genetic algorithm. Furthermore, we provide the improved crossover operations and mutation operation, including 1-parent crossover, 2-parent crossover, and mutation, based on the characteristic of the patent map systems. Compare to the traditional method, our proposed method reduces almost 50% time cost. Hence, our proposed method is quite suitable for online public media, such as world wide web. According to our proposed designs, we develop a prototype of the patent map system. Our prototype is designed by a popular web programming, PHP, and thus, it can be easily ported to platforms of different operating systems. We also present a case study of applying our system prototype to an industry-academy cooperation project. We analyze patent relationship among other related patents and make some patent evasion strategies for their current products. Through the industry-academy cooperation project, the product designers of Winton Company can have more different business strategies for benefiting their products from the proposed patent map system. Our future work will continue to study the impact of different GA-based operations (i.e., crossover and mutation) to the

quality of generated patent maps. More experiments will be conducted to verify the best design of GA-based operations. In addition, we continue to revise the website design based on the master thesis [21] to increase use experience on our system [17]. Currently, our prototype is used in Prof. Rain Chen's patent courses. The human-server interface will be improved according to the students' feedback.

Acknowledgment

This work was partially supported by National Science Council of Taiwan and National Cheng Kung University under grants NSC100-2221-E-006-267-MY2, NSC100-2221-E-218-049, and D101-13103. Authors thank Prof. Ding-Chau Wang for his contributions on partial design and maintenance of the web site.

References

- [1] M. M. Bronstein, A. M. Bronstein, R. Kimmel, and I. Yavneh. Multigrid multidimensional scaling. *Numerical Linear Algebra with Applications*, 13:23, 2006.
- [2] Claudia Canongia, Adelaide Antunes, and Maria de Nazare' Freitas Pereira. Technological foresight - the use of biotechnology in the development of new drugs against breast cancer. *Technovation*, 24:10, 2004.
- [3] Alfred Chen and Rain Chen. Design patent map: An innovative measure for corporative design strategies. *Engineering Management Journal*, 19:15, 2007.
- [4] Alfred Chen and Rain Chen. A seven-track protection mechanism for design knowledge. *IIGSS Academic Publisher Scientific Inquiry*, 8:11, 2007.

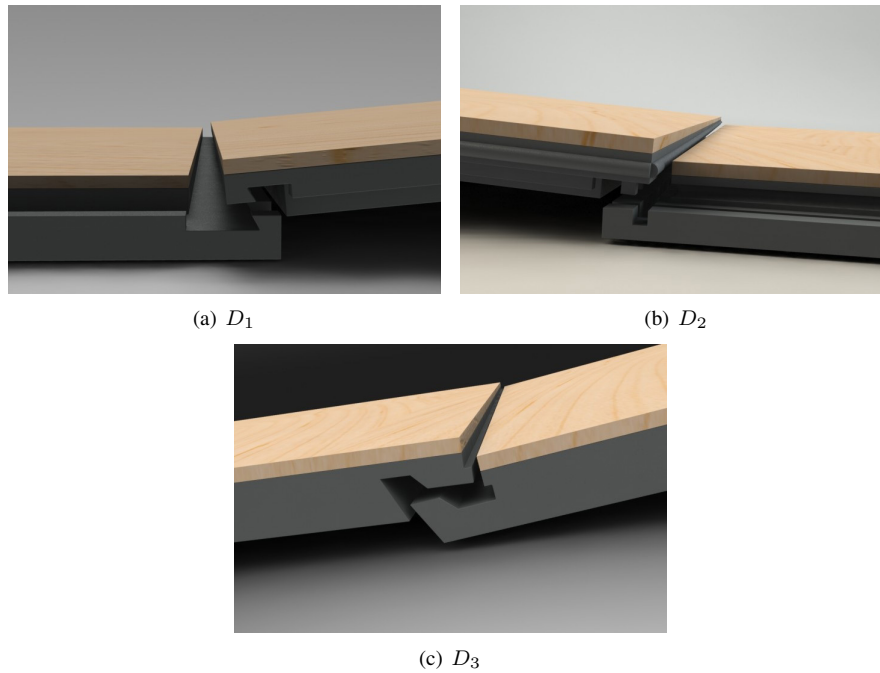


Figure 15: Three products of ground mats for the case study.

	1	2	3	4	5	6	7	8
1	0	4/0.810	4/0.880	4/0.500	4/0.630	4/0.750	4/0.810	4/1.000
2	4/0.810	0	4/0.940	4/0.750	4/0.810	4/1.000	4/0.810	4/0.440
3	4/0.880	4/0.940	0	4/0.750	4/0.810	4/1.000	4/1.000	4/1.000
4	4/0.500	4/0.750	4/0.750	0	4/0.500	4/1.000	4/0.880	4/0.810
5	4/0.630	4/0.810	4/0.810	4/0.500	0	4/1.000	4/0.560	4/0.940
6	4/0.750	4/1.000	4/1.000	4/1.000	4/1.000	0	4/1.000	4/1.000
7	4/0.810	4/0.810	4/1.000	4/0.880	4/0.560	4/1.000	0	4/0.630
8	4/1.000	4/0.440	4/1.000	4/0.810	4/0.940	4/1.000	4/0.630	0

Figure 16: Dissimilarity matrix of 3 products and 5 patents.

- [5] Chao-Chun Chen, Rain Chen, Ding-Chau Wang, and Ting-Ting Dai. Ga-based dissimilarity visualization engine for design patent map systems. In *Proceedings of the Eleventh International Conference on Hybrid Intelligent Systems (HIS 2011)*, pages 595–600, Malacca, Malaysia, December 05–08 2011.
- [6] J. L. Chen, S. J. Liu, and C. H. Tseng. Technological innovation and strategy adaptation in the product life cycle. *Technology Management: Strategies and Application*, 5:19, 2000.
- [7] Rain Chen. Design patent supported strategy: The core for the enhancement of corporate competitiveness. *National Digital Library of Theses and Dissertations in Taiwan*, no:122, 2004.
- [8] Rain Chen. Design patent map visualization display. *Expert Systems with Applications*, 36:12, 2009.
- [9] J. L. Herce. WIPO patent information service for developing countries. *World Patent Information*, 23:13, 2001.
- [10] S. H. Huang, H. R. Ke, and W. P. Yang. Structure clustering for chinese patent documents. *Expert Systems with Applications*, 34:7, 2008.
- [11] IBM SPSS Statistics. Statistical package for the social sciences (spss).
- [12] Syed A. Ahson Mohammad Ilyas, editor. *Cloud Computing and Software Service*. CRC Press, 2010.
- [13] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:27, 1964.
- [14] J.B. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29:15, 1964.
- [15] Shang-Jyh Liu and Joenson Shyu. Strategic planning for technology development with patent analysis. *International Journal of Technology Management*, 13:19, 1997.
- [16] A. Mina, Ronald Ramlogan, Gindo Tampubolon, and John Stanley Metcalfe. Mapping evolutionary trajectories: Applications to the growth and transformation of medical knowledge. *Research Policy*, 36:18, 2007.
- [17] B. Shneiderman. The next generation of graphical user interfaces: Information visualization and better window management. *Displays*, 17:5, 1997.

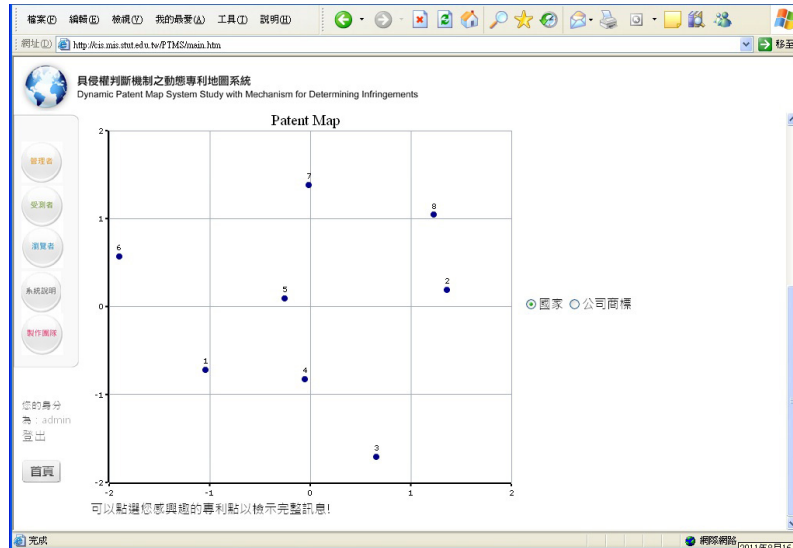


Figure 17: Patent map of 3 products and 5 patents.

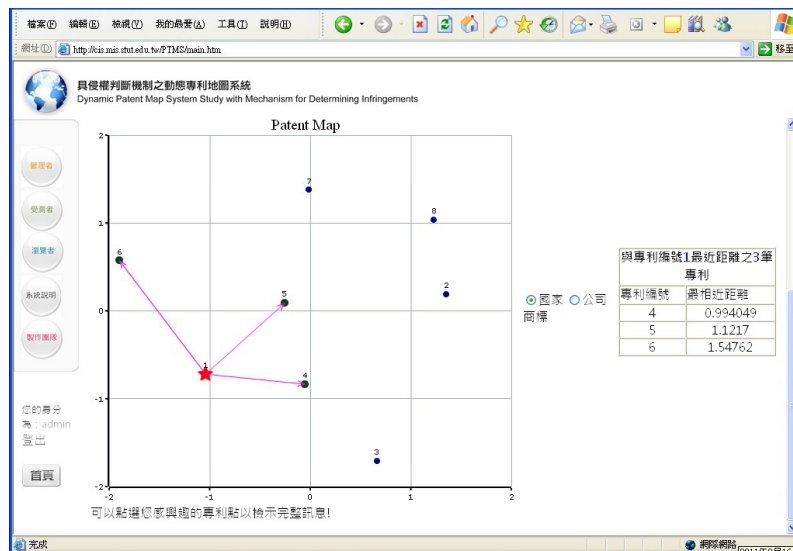


Figure 18: Query three nearest patents of D_1 .

- [18] Yuen-Hsien Tseng, Chi-Jen Lin, and Yu-I Lin. Text mining techniques for patent analysis. *Information Processing and Management*, 43:32, 2007.
- [19] USPTO United States Patent and Trademark Office. US patent law. <http://www.uspto.gov>, 2008.
- [20] Leo Wanner, Baeza-Yates Ricardo, Soren Brugmann, Joan Codina, Barrou Diallo, Enric Escorsa, Mark Giereth, Yiannis Kompatsiaris, Symeon Papadopoulos, Emanuele Pianta, Gemma Piella, Ingo Puhlmann, Gautam Rao, Martin Rotard, Pia Schoester, Luciano Serafini, and Vasiliki Zervaki. Towards content-oriented patent document processing. *World Patent Information*, 30:13, 2008.
- [21] Bei-Bei Yao. The research on patent map system and interface planning. *National Digital Library of Theses and Dissertations in Taiwan*, page 95, 2010.