Available online at www.ijo.ir

Iranian Journal of Optimization 1(2009) 151-161

Iranian Journal of Optimization

Integrating Case-Based Reasoning, Knowledge-Based Approach and TSP Algorithm for Minimum Tour Finding

Hossein Erfani*

Department of computer Lahijan I.A.U.

Abstract

Imagine you have traveled to an unfamiliar city. Before you start your daily tour around the city, you need to know a good route. In Network Theory (NT), this is the traveling salesman problem (TSP). A dynamic programming algorithm is often used for solving this problem. However, when the road network of the city is very complicated and dense, which is usually the case, it will take too long for the algorithm to find the shortest path. Furthermore, in reality, things are not as simple as those stated in AT. For instance, the cost of travel for the same part of the city at different times may not be the same. In this project, we have integrated TSP algorithm with AI knowledge-based approach and case-based reasoning in solving the problem. With this integration, knowledge about the geographical information and past cases are used to help TSP algorithm in finding a solution. This approach dramatically reduces the computation time required for minimum tour finding.

Keywords: Case-based reasoning, Knowledge-based approach, Traveling salesman problem, Case-based tour finder, Knowledge-based route finder, geographical information

* *E-mail: HErfani@gmail.com*

1 Introduction

One of the important computer applications in the tourism industry is minimum tour finding. Suppose a tourist rented a car and planned to drive around a city. He/she needs to know the way before going for one tour.

In Network Theory (NT), this minimum tour finding problem is called the traveling salesman problem [3]. A dynamic programming algorithm [2, 3] is normally used for solving this problem. However, when the road network is complex and dense, which is usually the case; it will take a long time for the algorithm to find the shortest path. But most of the paths searched by the algorithm are actually irrelevant because they cannot possibly be part of the solution. Consequently, it wastes a lot of computation time. In other words, it is usually not necessary to search the whole road network in order to find the shortest tour from one point to another. If the algorithm can incorporate some commonsense knowledge and knowledge about the geographical information of a particular place, those unnecessary searches can be avoided. Another consideration is that road networks in real situations are not as simple as those stated in NT. For instance, the cost of travel for the same part of a road network at different times may not be the same. In this project, TSP algorithm has been integrated with AI knowledge-based approach and case-based reasoning in solving the shortest tour finding problem. Knowledge-based approach [8] in our shortest tour finding context can be stated as using knowledge about the geographical information to prune the search space and to guide the problem solving. More specifically, knowledge tells TSP algorithm where it should search and where it should not.

Case-based reasoning [7, 8] is an AI technique which "remembers" previous problems and solutions and uses the knowledge of what worked before to solve the new problems. This is similar to our human experience. For example, if one wants to go to a place where he has been before, most probably he will not use the map to try to find a new route. Instead, he is likely to use the previous route in his memory to guide the present trip. If he wants to go to a place quite near to where he has been before, he will adapt (modify and extend) that old solution for the new problem. In section 3, we discuss in detail how case-based reasoning is of great help in route finding.

From our experience, we can say that TSP algorithm, AI knowledge-based approach and case-based reasoning complement one another very well in route finding. The advantages of each technique overcome the disadvantages of the others. This integrated approach dramatically reduces the computation time required in finding a tour.

The paper is organized as follows: in the next section, we discuss why each individual approach alone, i.e., TSP algorithm, knowledge-based technology and cased-based reasoning, is not suitable for realistic shortest tour funding. In section 3, their integration is presented as a good solution method. Section 4 describes the T-Finder system. Section 5 discusses the related work and section 6 concludes the paper. In the conclusion, it is also suggested that this integrated.

2 Why not use only one of these techniques?

A question that may be naturally asked will be "can we use only one of the techniques for solving the minimum tour finding problem?" In the following, we will analyze and explain why any single technique is not a good approach.

1. TSP algorithm is the only technique that is able to work independently to solve the problem. The algorithm searches in a weighted directed network as in Figure 1. It finds the shortest tour from a source (any node in a network) to the same node visiting specified nodes in the network. The weight for each edge in our shortest tour finding context could be distance or traveling cost between two adjacent nodes.

We will not discuss the algorithm in detail in this paper because it can be found in almost any book on algorithms or network theory.

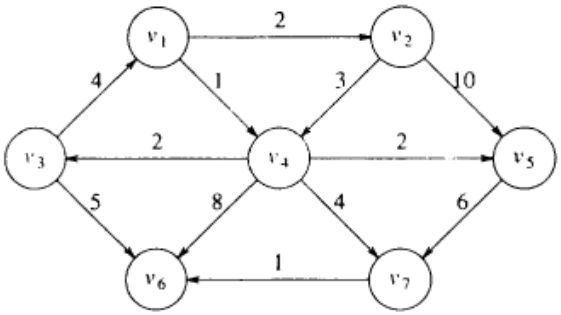

Figure 1. A weighted and directed network.

Although this algorithm is efficient; searching the whole road network with thousands of roads and junctions to find the shortest tour will still take a long time. As we mentioned before, it is also very wasteful in terms of computation. The reason is that it is not always necessary to search through the whole road network in order to find the solution. For example, if one is going from a central point to a southern point, most probably one will not consider those roads which lead to the north. Most of the time commonsense knowledge like this and geographical knowledge about sites can be used to guide us to find a shortest tour.

2. Using knowledge-based approach alone will not be efficient in solving the problem either. After applying commonsense knowledge and heuristic knowledge about the geographical situations, a search is still required to find the best solution. In most situations, knowledge alone is unable to guarantee the best solution. It can only isolate the part of the network which could contain the solution, or prune off part of the network that is unlikely to contain the solution. After that, a search algorithm has to be used to find the best solution in the isolated area. In this case, TSP algorithm comes to help because it is one of the most efficient algorithms for the problem.

3. Case-based reasoning means using old experiences to understand and solve new problems. A reasoner remembers a previous situation similar to the current one and uses that to solve the new problem. After the new problem is being solved, its solution will be stored as a new case, hence giving the system more familiar contents to solve problems. One of the key advantages of case-based reasoning is that it learns becoming more competent as it acquires new cases.

For our shortest tour finding problem, many shortest tours could be pre-stored. When the new problem is the same as an old one, the old shortest tour is retrieved for the new problem. When the new problem is similar to an old problem, the old solution could be adapted for the new problem.

However, using only case-based reasoning will not be sufficient. The first reason is that we cannot possibly store all the shortest tours from one point visiting any subset of other points for a huge network. If we cannot store everything, when there is no old case that exactly matches the new problem, adaptation is needed. However, complicated adaptation also takes time. Sometimes, adaptation may not even be possible as there is simply no similar old case in the case base.

The second reason is that even if this complete case base can be built, it still cannot handle situations when ad hoc problems such as traffic congestion and accident cause disruption to the road network. In these situations, we have to resort to searching. It is not possible for the case base to store all the solutions for abnormal road conditions.

From the above, it can be seen that any single technique will not be a suitable approach for shortest tour finding. In the next section, we will see that their integration provides an efficient solution.

3 Integrating TSP algorithm, knowledge based approach, and case based reasoning

From the above discussion we see that although each individual technique does not provide a good solution method for shortest tour finding, each of them has its advantages in solving the problem. Let us summarize both the known advantages and disadvantages.

3.1 Main advantages and disadvantages of each individual approach

The main advantages of each approach:

1. TSP algorithm is able to find the best solution.
2. Knowledge-based approach can isolate the part of the road network which is very likely to contain a solution.
3. Case-based reasoning gives the solution to the new problem instantly if there is a similar old case in the case base for the new problem.

The main disadvantages of each approach:

1. The major problem with TSP algorithm is that it does a blind search which can be time consuming and wasteful in terms of computation.

2. The problem with knowledge-based approach is that it is very hard to find the best solution without search.

3. The problem with case-based reasoning is that if there does not exist an old case in the case, base that can match the new problem, or that can be adapted for the new problem, it will not give any solution.

Analysis of the above advantages and disadvantages of each technique shows that they can easily complement each other. Thus, integrating them is the natural solution.

3.2 Integrating the three techniques

Obviously, in the integrated approach, we should keep the advantages of each individual technique, and at the same time use the advantage of one technique to overcome the disadvantages of the others.

As indicated before, the problem of blind search of TSP algorithm can be improved by incorporating commonsense knowledge and knowledge about the geographical situations. This knowledge helps to rule out those pans of the network which are unlikely to contain a solution.

Search space for TSP algorithm may also be pruned off by old cases stored for case-based reasoning. For example, in a situation where there is no exact case to solve a new problem, there may be an old tour which goes in the same direction but does not quite visit all nodes in the new case. Then TSP algorithm can be used to continue for the remaining part. Such a combined path, however, may not be the best solution. But in real life, a reasonably good solution will normally be sufficient.

Unlike TSP algorithm and knowledge which need to work together to solve a problem, case-based reasoning can solve some problems independently and also help TSP algorithm in some other situations (as just mentioned). For example, when there is an old case in the case base which matches the new query, there will be no need to search with TSP algorithm or to use any geographical knowledge, but simply output the tour of the old case. This is similar to our experience of going to a place where we have been before without referring to a map again. If there is no exact case in the case base which matches the new requirement, the case-based reasoner tries to find a similar case and adapt it for the new problem. In these two situations, search is not needed. Case-based reasoning can produce results effortlessly. However, when a similar case is not found, the case-based reasoner will not be able to provide a solution. Then, it will try to find a case which can provide a partial solution and leave the rest of the problem to TSP algorithm and knowledge. This links case-based reasoning and TSP algorithm. In the worst case, when even a partial solution cannot be found, it will pass the whole problem for search.

The above description of the integrated approach also suggests a sequence of problem solving in this approach. That is, the case-based reasoner should try to solve the problem first because it is more efficient. If the case-based reasoner fails, the problem will then be passed to TSP algorithm and knowledge-based approach. Another important thing that has to be decided in the integrated approach is how much work each subsystem should do. For example, if the case-based reasoning is allowed to do very complicated adaptation, it will be time consuming and the result obtained might not be good. In our system, the case-based reasoner is only allowed to do limited adaptation, which could be done with little effort but guarantee good solutions, and leave those difficult situations to TSP algorithm and knowledge.

4 The T-Finder system

Using the integrated approach, we may implement the system T-Finder for tour finding in Isfahan. Isfahan is a historical city state with a complex network of streets, roads and highways. A person traveling from one location to a new place needs to consult a street directory. Unfortunately, this is often a very time consuming process. It is also painstaking and frustrating as there is no indication of the shortest tour leading to the destination. It is often up to the driver to find his own way around by trial and error. The T-Finder system is built to provide help in these situations.

4.1 T-Finder system architecture

Figure 2 shows the architecture of T-Finder. A brief description of each module is given below. Detailed discussion on CBTF and KBTF will be given in the next two subsections.

Interface: Interface is for the user to issue his/her request (e.g. source or starting point and visiting places) or query and to report problems. The problems refer to traffic congestions, accidents, etc. If a problem has been reported, the system needs to modify the database. Interface also outputs the tour found to the user.

MANAGER: The whole system is controlled by MANAGER. When a query is issued or a problem reported, it dispatches jobs to various modules.

Database: It contains the map represented as a weighted network just as the network for TSP algorithm (Figure 1). The weight for each road could be distance or travel time required.

Database maintainer: It maintains the database. For example, when a road is congested, it will change the weight attached to the road. When the road is cleared, the normal weight will be restored. As another example, if the source or the visiting places is not a physical junction but a point on a road, an artificial junction is created for the source or the visiting places.

CBTF: This is the Case-Based Tour Finder module. In most situations, the MANAGER will ask CBTF to find a solution first because it is more efficient than KBTF (Knowledge-Based Tour Finder). If CBTF is unable to find a solution for a query, it will report to MANAGER which will then pass the problem to KBTF.

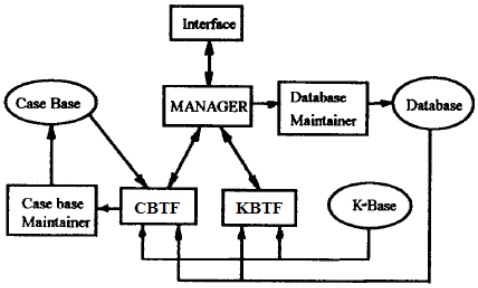

Figure 2.The T-Finder system architecture

Case base: It stores old cases.

Case base maintainer: It maintains the case base, such as adding new cases and removing old cases.

KBTF: This is the Knowledge-Based Tour Finder module. It uses knowledge in the knowledge base (Kbase) to prune off part of the network and then calls for TSP algorithm to find the best solution in the reduced network.

K-Base: This is the knowledge-base. It contains all the commonsense and geographical knowledge about Isfahan for KBTF and CBTF to make various judgments and decisions.

4.2 Case-based tour finder (CBTF)

Our case-based reasoning system has two major components: a case base and a problem solver. Each case in the case base consists of a list of junctions which is the solution path. In each case it is also indicated whether the case should be used for peak hours or off-peak hours traveling because certain places always encounter traffic congestions during peak hours. After MANAGER has issued a request (the source, visiting places and some other important features like time of travel, etc.), the system will try to find some similar cases in the case base to solve the problem. We shall use a small piece of the Isfahan map (Figure 3) to illustrate. Note that we use very simple cases in the examples below for easy explanation. In T-Finder, cases can be of any

size. Due to the space limitation, only one small piece of the map is used for all examples. Hence, cases and solutions in each example should be seen as independent of each other.

1. Total match

If both the source and visiting places of the new problem are found in an old solution in the case base, we will extract this part of the solution and output to the user. This operation can be done effortlessly.

E.g., Problem: Source=2, Visiting places=36 27 68 43, A matched case in the case base:

(2, 68 55 43 27 19 36)

Method: The part in the case which is relevant to the new problem is extracted Output: (2, 68 43 27 36)

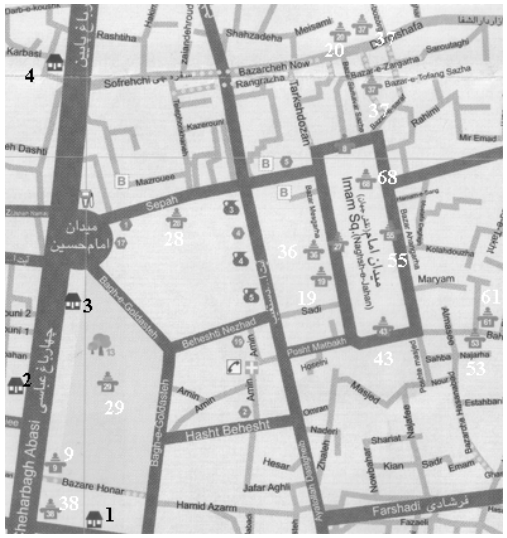

Figure 3.A sample map

2. Finding a nearby junction of source or visiting places

When an exact matching case is not found in the case base that involves both source and visiting places, adaptation is required. However, this adaptation is limited to only nearby junctions. If a solution can be found by this simple adaptation, the computation required is also very small.

E.g., Problem: Source= 2, Visiting places = 20 28 37, Relevant cases in case base:

(1, 20 28 37) and (3, 28 37 20) Method: Since there is no case in the case base that involves both the source and visiting places, the system looks for the nearby sources of the original source 2 (i.e. 1,3) in the database.

Since the shortest path between the source and every visiting place goes through source 3, but not through source 1 we will use the second case for new case. Output: (2, 28 37 20)

3.Finding part of the solution for the problem and the remaining sub problem is passed over to KBTF.

This strategy is used when the above two fail. The system will try to find a relatively straight path leading from the source to visiting places in the case base. If it can extract part of the path that satisfies a set of constraints (which makes it a good partial solution), it will pass the partial solution and the rest of the

problem over to KBTF through MANAGER. The set of constraints is stored in the knowledge base and it varies according to different geographical situations.

Currently, the system passes 3 different partial solutions (if possible) to KBTF, which will extend these partial solutions to find the best complete solution. Obviously, this will be better than passing only one partial solution to KBTF.

We limit the adaptation strategies to the simple ones because as mentioned before, complicated adaptation may take a long time which defeats the advantage of using case-based reasoning. It is also dangerous to do complicated adaptation (such as merging of a few cases) because it is difficult to guarantee a good final solution. When CBTF cannot find a solution, it will inform the MANAGER, which will pass the problem (or partial problem) to KBTF.

In a typical case-based reasoning system, when new solutions for new problems are produced, they will normally be stored in the case base. This increases the problem solving power of the system. In our situation, T-Finder does not store any new solution which comes from adaptation of similar cases because we believe that these adaptations are easily done. If all the cases are stored, we will have storage problem. Thus, we need to be selective, which means that the system only stores those important solutions (this is done by the case base maintainer) that cannot be easily adapted from existing cases.

Whenever CBTF reports a failure, it will pass over to KBTF to search for the optimal path. The case base maintainer will use this solution to update the existing case base. It does the maintenance as follows: the new solution is first checked against the cases in the existing ease base for any cases which are subsets of the new solution. If such cases are found, they will be deleted and only the new case will be stored.

4.3 Knowledge-based route finder (KBTF)

When CBTF cannot find a solution for a new problem, it will be passed to KBTF. In the situation when there is an emergency, such as accident or traffic congestion, the MANAGER will pass the problem to KBTF directly as CBTF cannot cater to such ad hoc situations.

The most important point in KBTF is to use commonsense knowledge and knowledge about the geographical situations to prune off part of the network which is unlikely to contain the solution. This is done in KBTF by using various grid combinations to isolate and to mark the part of the road network that needs to be searched. This marked part of the network is then fed to TSP algorithm to search for the best solution.

The job of the knowledge is to decide the combination of grids for each problem according to the locations of the source and visiting places, and the road network and/or other geographical objects in between. Due to space limitation, let us use only two examples to illustrate the point.

Example 1. The source and the visiting places are all in the city center. In the city center the road network is very dense and there is no natural obstacle. So the grid is a rectangular as in Figure 4.

After the grid has been decided with knowledge, the system will mark the road network inside the rectangular grid. There are two purposes for doing this. Firstly, it isolates the part of the network for TSP algorithm to search. Secondly, it makes sure that the source and the visiting places are connected within the grid since knowledge cannot guarantee this connection. If it is found that they are not connected, the original grid will be increased so as to cover a bigger area. When the isolated network is marked, it is sent to TSP algorithm to search for the best solution.

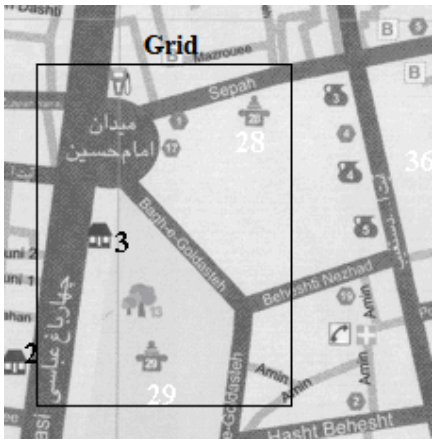

Figure 4 . A sample grid far a part of the city center

Example 2. Traveling over a long distance. For long distance traveling, drivers normally would like to take major roads rather than minor roads. In this case, we will need to have a combination of grids (rather than just one) and to differentiate the major and minor roads. In certain grids we need to include the minor as well as major roads, while in others we will only have the major roads.

Let us assume the source and visiting places are all on some minor roads. In this case, we need to have a grid at the source which include the minor roads and also the major roads so that they can be linked. At the visiting places, some grids are needed which include the major as well as the minor roads. However, in the middle region, only the major roads need to be included. This prunes off a big portion of the search space. See the diagram below (Figure 5), the source is in one of the small town areas and the visiting places are in the city center.

The size of each grid is decided according to different geographical situations. After the grids have been decided, as in example l, all the roads (major and minor) in grids 1 and 2 will be marked. But in the region outside of 1 and 2 but inside 3, only the major roads will be marked. The marked road network is given to TSP algorithm to search.

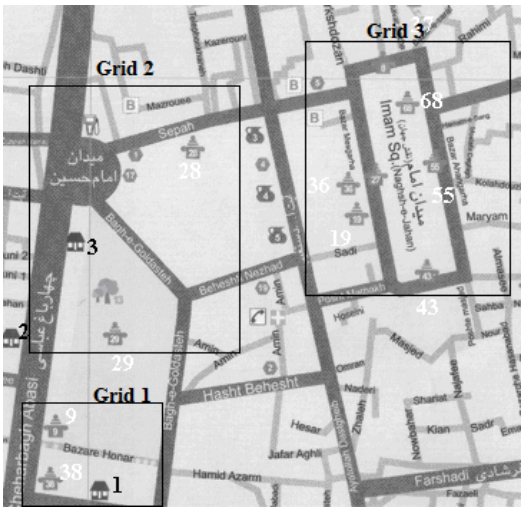

Figure 5.A sample grid combination for a long distance travel

After a solution has been found, it is passed back to MANAGER which will output it to the user through the interface. MANAGER also sends the solution to the case base maintainer for case base updating. Since the roads in the grids are marked, the system needs to unmark them after the search.

5 Related work

Although the traveling salesman problem [2, 3] has been studied for decades and good algorithms are available, in our literature survey, we did not find any report on knowledge-based approach to the problem. We think that one of the reasons could be that TSP algorithm or others are already very efficient for the task. The knowledge-based approach is mostly used for solving problems that do not have efficient algorithmic approaches [1, 4, 10]. However, from our experiments, we can safely conclude that although TSP algorithm is efficient, knowledge-based approach can be successfully incorporated for an even more efficient solution method for route finding.

To the best of our knowledge, there is also no work done in using cased-based reasoning for TSP, although there are case-based reasoning applications in many areas [e.g., 5, 7, 9], e.g., design, planning, diagnosis, explanation, etc. We believe that case-based reasoning is very important for TSP. It is also potentially useful for any geographical information system.

6 Conclusion and future work

This paper presents an integrated approach for solving traveling salesman problem. This approach consists of TSP algorithm, knowledge-based technique and case-based reasoning. From our experience, we can say that the three techniques integrate and complement one another very well. They together provide an efficient solution for traveling salesman problem.

Currently, I plan to develop the system in Isfahan (if the financial sources are provided) which will be useful for public and tourists, who use public transport. Finally, I would like to make some comments on geographical information systems. T-Finder is one such system. I believe the combination of knowledge-based approach, case-based reasoning and possibly database techniques and theoretical methods, such as those in databases, operations research and network theory (in our case, it is Traveling Salesman Problem algorithm) are very important for any realistic geographical information system. One of the most important characteristics of a geographical database is its huge size. Clearly knowledge-based approach is essential for

pruning the search space in query processing. Case-based reasoning is also very useful because it can save the solutions for old queries. When similar queries are issued again later, the old solutions can be retrieved and output straightway. If the old solutions are not quite suitable for the new problems, they may be adapted. In these situations, there is no need to search, which is typically computational intensive and time consuming.

References

[1] Brownston L., Farrell R., Kant E. and Martin N., Programming Expert Systems in OPSS, Addison-Wesley Publishing Company, 1985.

[2] Neapolitan R., and Naimipour K., Foundations of Algorithms Using C++ Pseudocode, Third Edition ,Jones and Bartlett Publishers, 2004.

[3] Chen W., Net Theory And ITS Applications Flows in Networks World Scientific Publishing ,2003.

[4] Stulp F., A Knowledge-Based Algorithm for the Internet Transmission Control Protocol (TCP), Bulletin of Economic Research, 54, 69-94, 2002

[5] Henessy D., and Hinkie D., Applying Case-Based Reasoning to Auto Clave Loading, IEEE EXPERT, 21-26, 1992.

[6] Pearce M., Goel K. A. and Kolondner J. L., Case-Based Design Support--A case study in Architectural, IEEE EXPERT, 14-20, 1992.

[7] Kolodner F. L., Improving Human Decision Making Through Case-Based Decision Aiding, AL Magazine, Vol. 12 (2), 52-68, 1991.

[8] Aamodt A. and Plaza E., Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches AICom - Artificial Intelligence Communications, IOS Press, 7(1), 39-59, 1994.

[9] Simoudis E., Using Case-Based Retrieval for Customer Technical Support, IEEE EXPERT, 7-11, 1992.

[10] Tsai M., Sung C., Cheng-Wei Lee; Shih-Hung Wu; Chorng-Shyong Ong; Wen-Lian Hsu, A knowledge-based approach to citation extraction,Information Reuse and Integration, Conf, 2005. IRI -2005 IEEE International Conference on.15-17 Aug. 2005, 50 – 55.

[11] Cheetham W., and Goebel K., Appliance Call Center: A Successful Mixed-Initiative Case Study, Artificial Intelligence Magazine, Volume 28(2), 89-100, 2007.

[12] Topel T., Neumann J., and Hofestadt R., A Medical Case-Based Reasoning Component for the Rare Metabolic Diseases Database Ramedis, Computer-Based Medical Systems, CBMS apos. Twentieth IEEE International Symposium on Volume, Issue, 20-22, 7 – 11, 2007.