

# Towards a Multi Objective Path Optimisation for Car Navigation

Ching-Sheng Chiu<sup>1</sup> and Chris Rizos<sup>2</sup>

<sup>1</sup>*(Department of Urban Planning & Spatial Information, Feng Chia University)*

<sup>2</sup>*(School of Surveying & Spatial Information Systems, The University of  
New South Wales)*

(E-mail:cschiu@fcu.edu.tw)

In a car navigation system the conventional information used to guide drivers in selecting their driving routes typically considers only one criterion, usually the Shortest Distance Path (SDP). However, drivers may apply multiple criteria to decide their driving routes. In this paper, possible route selection criteria together with a Multi Objective Path Optimisation (MOPO) model and algorithms for solving the MOPO problem are proposed. Three types of decision criteria were used to present the characteristics of the proposed model. They relate to the cumulative SDP, passed intersections (Least Node Path–LNP) and number of turns (Minimum Turn Path–MTP). A two-step technique which incorporates shortest path algorithms for solving the MOPO problem was tested. To demonstrate the advantage that the MOPO model provides drivers to assist in route selection, several empirical studies were conducted using two real road networks with different roadway types. With the aid of a Geographic Information System (GIS), drivers can easily and quickly obtain the optimal paths of the MOPO problem, despite the fact that these paths are highly complex and difficult to solve manually.

## KEY WORDS

1. Shortest path.    2. Least node.    3. Minimum turn.    4. Car navigation each.

1. INTRODUCTION. Contemporary car navigation systems generally provide the Shortest Distance Path (SDP) between two locations—one of the most important functions for drivers. The key technology underpinning SDP is the Shortest Path (SP) algorithm. The SP problem was proposed in the 1950s and has been intensively studied in the intervening years. Many algorithms for solving the SP problem have been developed (e.g. Moore, 1957; Bellman, 1958; Dijkstra, 1959). Although selecting driving routes with the assistance of SDP information is easy to understand and relatively intuitive, as several researchers have pointed out, drivers may apply multiple criteria to decide their driving routes (e.g. Hansen, 1980; Climaco and Martins, 1982; Henig, 1985; Hallam et al., 2001; Chakraborty et al., 2005). Therefore the single-objective SP problem was extended with two objectives. Usually, the two objectives are the travel time and cost. Nevertheless, in addition to distance, time and cost objectives, there are other objectives that may be relevant to drivers.

For example, drivers may prefer a road with a higher speed limit. They may also be concerned with how many intersections or traffic signals they will pass, and/or how many turns they have to make until they reach their destination. Hence, to develop a more precise simulation of the decision-making behaviour of drivers for route selection, it is necessary to extend the SP problem with two or more objectives.

From the Multi Objective Programming (MOP) literature, a multi objective SP problem can be classified as a MOP problem and will share the same problem-solving procedures as the MOP problem. Unlike identifying the optimal path of the SP problem, which could proceed directly to an optimal solution, a two-step process is required to identify the optimal solution of a MOP problem. The first step is to find all non-inferior (non-dominant or Pareto-optimal) solutions that are feasible solutions to the MOP problem, if no other feasible solutions exist that will yield an improvement in one objective without causing degradation in at least one other objective. The second step tends to emphasise the linkages of decision-makers and their preferences through which the best-compromise solution can be identified from non-inferior solutions (Cohon, 1978).

In the past, researchers dealing with SP problems of a network were focused on the single objective. Even when the SP problems with two objectives were proposed, they were solved directly by the Dijkstra's algorithm. Since the SP problem with two objectives is a type of MOP problem, solving it by the integration of the Dijkstra's algorithm and MOP problem-solving techniques is more meaningful. However, as the links of a network increase, the decision variables also increase. Hence MOP problem-solving techniques were rarely applied in solving the multi objective SP problem. If a route selection problem contains more than one type of decision variable, solving it is much more complex than solving the multi objective SP problem. This is due to the Dijkstra's algorithm being only applicable for solving the SP problem with links as decision variables. To solve problems with nodes and two-links as decision variables, new algorithms have to be developed.

The remainder of this paper is organised as follows. A Multi Objective Path Optimisation (MOPO) model is proposed in Section 2. Firstly, utilising the MOP approach, five objectives corresponding to three types of decision variables (i.e. nodes, links and turns) are taken into account in the construction of such a MOPO model and each is a Single Objective Path Optimisation (SOPO) model. Section 3 presents the development of techniques and algorithms for solving the MOPO and SOPO problems. Firstly, the issue of choosing proper techniques that are applicable for solving the MOPO problem is addressed. Secondly, the development of algorithms for solving the Least Node Path (LNP) and Minimum Turn Path (MTP) problems is presented. In Section 4, based on the MOPO and SOPO problems, experimental programs for solving the problem with nodes, links and turns were developed and implemented by integration with a commercial Geographic Information System (GIS) package. To demonstrate the advantages of the MOPO model for supporting more diverse and richer information to drivers to assist route selection, several experiments were conducted, utilising two real road networks with different roadway types and numbers of nodes and links. The conclusion ends the paper.

## 2. MULTI OBJECTIVE PATH OPTIMISATION MODEL.

Identifying objectives and decision variables is the first step in modelling a MOPO

problem. In addition to the shortest distance objective, two groups of objectives - *aggregate* and *disaggregate* decision criteria - can be used to establish the MOPO decision model. The aggregate decision criterion stands for those objectives where some of their decision variables are not strongly related to or cannot be directly measured by nodes, links, or turns. The disaggregate decision criterion is just the opposite. Objectives of the former group (including time, cost, comfort, and safety) are used to evaluate the influence on the MOPO problem from decision variables like traffic condition, automobile models, etc., that are assumed the same for all roadway networks. On the other hand, objectives of the latter group (including passed intersections, cumulative distance, roadway capacity, roadway grade, and number of turns) are used to represent the individual influence on the MOPO problem from the decision variables such as nodes, links, or turns.

Constructing a *hierarchical value tree* is the best way to represent the MOPO problem. For *aggregate* and *disaggregate* decision criteria, one can construct the tree as a two-level hierarchy where the upper-level contains aggregate criteria and the lower-level includes disaggregate criteria. Perhaps one could also organise the tree on only one level including all criteria. In principle, the *travel time* is affected not only by how many intersections are passed and the distance to be travelled, but also by traffic conditions, and many other factors. Similarly, *travel cost* is affected not only by the distance to be travelled, but also by the engine's volume, petrol prices, etc. Since (in contrast to passed intersections and distance objectives) the traffic conditions and petrol prices are relatively difficult to measure, they are not applicable for modelling the problem within a car navigation system. In addition, because both comfort and safety criteria contain many decision variables that cannot easily and directly be measured, they also are not applicable to be taken into account as the decision criteria for the MOPO problem.

Although a hierarchical value tree is a commonly used approach to describe a complex decision-making problem, constructing one for a MOPO problem may not be necessary. The reason is quite simple—the value trees constructed by decision-makers vary from one to the other. If drivers had to spend a lot of time constructing a value tree for a MOPO problem before they could make a decision, it would not be applicable to a practical car navigation system. To drivers, a value tree with only one level is perhaps the best way to describe the MOPO problem, due to its simplicity and convenience. On the other hand, for the MOPO problem, even a value tree with only one level may be too complex for drivers to make decisions if there are too many decision criteria. Therefore, a smaller set of decision criteria makes the MOPO more applicable to practical use within a car navigation system. In this paper only the disaggregate objectives are taken into account to establish a decision-making model for MOPO problems.

Let  $O_i$  denote the multi objective function with minimisation goals. Therefore the MOPO problem can be formulated as:

$$\text{Min } O_i[f_1(\text{Intersection}), f_2(\text{Distance}), f_3(\text{Grade}), 1/f_4(\text{Capacity}), f_5(\text{Turn})] \quad (2.1)$$

Note that all optimisation functions of the MOPO problem can be solved individually or in various combinations. The former approach assumes that each optimisation function is independent or has no relation to the others, while the latter does not. Moreover, if the latter approach is applied, there are many combinations which can be used (e.g. the combination of  $f_1$  and  $f_2$  or  $f_1, f_2$  and  $f_3$ , etc.). Also note the

different decision variables used by each SOPO problem of the MOPO model. For the passed intersections objective, the decision variable is a *node*. For cumulative distance, roadway grade and roadway capacity, the decision variable is composed of two nodes or a *link*. For the turn objective, the decision variable is measured with three nodes or two links. It is also worth noting that, except for the roadway capacity objective, all objectives in the MOPO problem are pursuing a minimisation goal.

2.1. *Measuring Objectives with Links as Decision Variables.* There are three objectives in this group, including *cumulative distance*, *roadway grade* and *roadway capacity*. The first objective in this group is *cumulative distance*. In practice, the distance is measured by the actual length of links (in units such as metres). The second objective in this group is *roadway grade*. Roadways with a higher grade promise to have greater turning radii, higher quality pavement, and wider lanes. In contrast with lower grade highways, it also implies that it is much safer, more comfortable, and can be driven at a higher speed. Moreover, driving at a higher speed implies lower fuel consumption. Therefore, driving on higher grade highways is the ultimate goal of drivers when they are planning their automobile travel. It is worth noting that the decision variable of roadway grade can be represented by an ascending or descending integer value. The goal for the grade objective should be minimisation if the former method is applied.

The last objective in this group is *roadway capacity*. The lane width of highways or streets is designed and deployed usually according to the roadway grade. The higher grade roadways will have wider lanes. Wider lanes in a roadway network imply that drivers can drive their automobiles in a relaxed manner and more easily than on narrower lanes. Accordingly, more lanes also promise higher capacity, and easier and safer passing of one automobile by another. Imagine how much more dangerous it is to overtake the front vehicles on a single-lane highway than on a highway with two lanes travelling in the same direction. Hence, the ultimate benefit for drivers driving on roadway networks is maximised capacity.

All the objectives mentioned above can be formulated by applying the same linear integer programming form as follows:

$$\text{Minimise } f_2, f_3 = \sum_{(i,j) \in E} e_{(i,j)} w_{(i,j)} \quad (2.2)$$

$$\text{Minimise } 1/f_4 = 1 / \sum_{(i,j) \in E} e_{(i,j)} w_{(i,j)} \quad (2.3)$$

$$\text{subject to } \sum_{(j,i) \in E} e_{(j,i)} - \sum_{(i,j) \in E} e_{(i,j)} = \begin{cases} 1 & \text{if } i = O \\ 0 & \text{for all } i \in N - \{O - D\} \\ -1 & \text{if } i = D \end{cases}$$

$$e_{(i,j)} = [0, 1] \quad \text{or all } (i, j) \in E$$

where  $w_{(i,j)}$  can be substituted with the distance, and roadway grade or roadway capacity to represent each corresponding objective. Note the difference between Equation (2.2) and (2.3). Because the capacity objective function pursues the maximum value as a goal, the same result can be obtained by converting the original function into a reciprocal form.

2.2. *Measuring the Passed Intersections.* Intersections are the main bottlenecks in urban street networks. Even in highway systems, intersections can also cause some traffic delay regardless of whether the intersections are provided with signals or not. Going through intersections forces vehicles to slow down, or fully stop, in order to safely pass intersections where traffic facilities are installed to control the traffic flow. Even if on a highway or express road, the speed limit at the interchange is slower than on the major lanes. When vehicles are merging from the interchanges to the major lanes, the traffic flow is also delayed. It is sufficient to conclude that the more intersections a path contains, the greater the resulting delay, despite the intersections being provided with signals or not.

Although the traffic delay caused by intersections can be measured by the travel time, there are many other aspects remaining of notable significance to drivers. The traffic light is the major tool to control and maintain traffic flows at intersections, regardless of the location on highways or street networks. However, a path associated with more traffic lights increases the probability that drivers will be forced to stop and wait. The resulting increase in acceleration events (from a stopped to moving status) also implies that more fuel will be consumed. Accordingly, during the acceleration process, drivers and passengers suffer and feel uncomfortable from the change in inertia. Moreover, the repeated operations associated with the increased number of stops, and increased complexity of the driving task will tire drivers. Therefore, it is reasonable to assume that minimising the number of passed intersections for a trip is the preference for all drivers and passengers. For the path optimisation problem with the objective function of minimising passed intersections, the tree of a directed least node path from a given origin node to destination node can be formulated as:

$$\begin{aligned} \text{Minimise } f_1 &= \sum_{i \in N} i & (2.4) \\ \text{subject to } \sum_{i=1}^n i &\leq n \\ \sum_{i=1}^n i &> 0 \\ i &= \begin{cases} 1 & \text{if } i \text{ is selected} \\ 0 & \text{otherwise} \end{cases} \text{ for all } i \in N \end{aligned}$$

2.3. *Measuring the Number of Turns.* Making *turns* implies that drivers are forced to slow down and pay more attention to driving. Turns are usually made at road intersections. In contrast with driving on roadways, passing an intersection implies that drivers will encounter more conflict and potential crash points – increasing traffic-accident probabilities, and decreasing safety. On the other hand, since traffic conditions at roadway intersections are very complicated, drivers are forced to focus more on driving and to take care of any situations that may arise. This increases the driving stress on drivers and may result in the drivers' fatigue. Furthermore, turns may also make both drivers and passengers suffer nausea caused by centrifugal force. Hence, it is acceptable for most drivers to set a minimisation objective for making turns when they plan their automobile travel. Therefore, for the path optimisation problem of the objective function of a minimum number of turns, the tree of a directed minimum-turn path from a given origin node to destination node

can be formulated as follows:

$$\begin{aligned}
 \text{Minimise } f_5 &= \sum_{(i,j),(j,k) \in E} e_{(i,j,k)} & (2.5) \\
 \text{subject to } \sum_{j|(i,j) \in E} e_{(i,j)} - \sum_{j|(j,i) \in E} e_{(j,i)} &= \begin{cases} 1 & \text{if } i = O \\ 0 & \text{for all } i \in N - \{O, D\} \\ -1 & \text{if } i = D \end{cases} \\
 \sum_{k|(j,k) \in E} e_{(j,k)} - \sum_{k|(k,j) \in E} e_{(k,j)} &= \begin{cases} 1 & \text{if } j = O \\ 0 & \text{for all } j \in N - \{O, D\} \\ -1 & \text{if } j = D \end{cases} \\
 e_{(i,j)} \text{ and } e_{(j,k)} &= \{0, 1\} \text{ for all } (i,j) \text{ and } (j,k) \in E
 \end{aligned}$$

**3. SOLVING THE MOPO PROBLEM.** Unlike identifying the optimal solution of the single-objective problem, it takes two steps to identify the best solution for a MOP problem. The first step focuses on employing the Dijkstra algorithm to identify the feasible solutions for the *cumulative distance*. The second step addresses the challenge of choosing appropriate techniques that are applicable for solving the MOPO problem and the effort of investing to obtain non-inferior solutions and best-compromise solution. Nevertheless, the Dijkstra algorithm is not applicable for solving both *passed intersections* and *number of turns*. To solve these two problems, two new algorithms will be presented in this section.

**3.1. MOP Techniques for Solving the MOPO Problem.** The decision-maker's preferences and the number of decision-makers are two important characteristics used to establish the decision-making process (Brownlow and Watson, 1987). The former indicates when the preferences will be given by the decision-maker, while the latter defines how many decision-makers will join the decision-making process. The number of decision-makers referred to above is actually two sets of problems, namely the *single-decision-maker problem* and the *multiple-decision-maker problem*. The former includes those situations in which there is a single-decision-maker or group-decision-makers that share similar objectives and preferences. The latter situation is directed at those cases in which there are many decision-makers or interest groups, each of which has its own conflicting objectives. In the case of the MOPO problem applied to car navigation, there is only one driver at a time who will play the role of decision-maker. The discussion of which techniques are more applicable for solving the MOP problem will focus on the single-decision-maker case.

The key to solving the single-decision-maker MOP problem is determining when and how to obtain the preferences from the decision-maker. There have been many attempts to classify techniques for solving the MOP problem by utilising the above two criteria. Among which the preference-flow concept proposed by Cohon (1978) suggests a strong relationship between the decision-maker and the analysts, which respectively stand for who articulates preferences and techniques for solving the MOP problem. In other words, the preference-articulation time, which refers to when the preferences are to be given, before, during or after finding all non-inferior solutions, may give more importance to the role that a decision-maker may play (Hwang and



Lin, 1987). Based on Cohon (1978), the preference-flow classification categorises the technique into three types:

- *Type I -No preference articulation.* Preferences are implicitly articulated by decision-makers after an approximate or exact non-inferior set is generated. Methods of this type includes the  $\epsilon$ -constraint (Haimes, 1973), *global criterion method* (Salukvadze, 1974), and *weighted-sum method* (Zadeh, 1963).
- *Type II -Prior preference articulation.* Either certain desired achievable objectives or definite pre-ordering goals can be performed by decision-makers prior to finding the non-inferior solutions. Methods in this group include *goal programming* (Charnes et al., 1969), *multi-attribute utility function* (Keeney and Raiffa, 1976; Berger, 1980), and *lexicographic* (Ben-Tal, 1980; Rao, 1996; Sarma et al., 1993).
- *Type III -Interactive or progressive preference articulation.* Methods in this group include: *probabilistic trade-offs development method* (Goicoechea et al., 1979), *step method* (Benayoun et al., 1971), and *sequential multi objective problem solving method* (Monarchi et al., 1973; Goicoechea et al., 1982).

It is clear that applying techniques in Type III to identify the best-compromise solution to the MOPO problem is not a wise idea. This is not only because of the complex burden of the driver interactively participating in the process of identifying best-compromise solutions requires repeated iterations, but also because it is hard for the driver to sharply or clearly distinguish differences in objectives and alternatives. The complexity of articulating preferences and the computational efficiency for obtaining non-inferior solutions are important criteria to evaluate which methods based on Type I and II techniques are most applicable for solving the MOPO problem. Undoubtedly, for car drivers to specify the range of feasible bounds for each objective is very easy. However, specifying meaningful ranges of feasible bounds that result in the MOPO problem having an optimal solution is very hard for drivers in practice. Imagine how difficult it is for drivers to judge what is the reasonable distance and travel time between two places, or how many intersections will be passed and turns to be made from an origin to a destination. It is hard for drivers to specify any target values for objectives of the MOPO problem. This is because the decision-maker may not know whether a path is valid or not, let alone what is the best value. Hence, the  $\epsilon$ -constraint, multi-attribute utility function and global criterion methods for solving the MOPO problem are not wise choices.

There are two main advantages that make the lexicographic method highly applicable for solving the MOPO problem. Firstly, it is due to the simplicity in requiring a decision-maker to rank the objectives in order based on one's preferences from best to worst. Once the order of the objectives has been obtained, the optimum solution can easily be identified by starting with the most important objective and proceeding according to the assigned order of the objectives. Secondly, even if a decision-maker does not provide ranking information for the objectives, it is also possible to select randomly an objective to be optimised at each time.

The weighted-sum method has considerable advantages due to its simplicity. Since there is only one driver who will play the role of decision-maker, the MOPO problem can be classified as the single-decision-maker problem in which the relative importance among objectives is defined only by one person. For every driver,

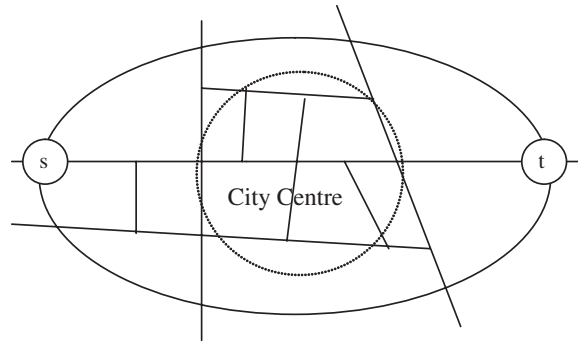


Figure 1. An example of a radial-circumferential network.

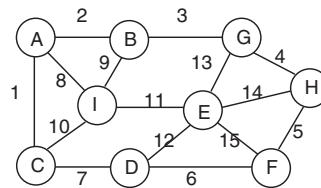


Figure 2. An example network of the LNP problem.

articulating one's preference in order of relative importance of objectives is easily done.

3.2. *Solving the Least Node Path Problem.* Although the SDP between two locations as determined by SP algorithms provides useful information for selecting driving routes, does this method conclusively result in the optimal path in a real road network, typical of the street networks in an urban area? The answer is very likely "No". If the optimal path is measured by travel time, it is obvious that it might not be the SDP. This is because most traffic delays occur at traffic junctions. Take a radial-circumferential network as an example, as depicted in Figure 1. If the start node  $s$  and destination node  $t$  are located at both sides of a radial-circumferential network, clearly the shortest distance path might go through the city centre along the straight line connecting  $s$  and  $t$ . However, since the total passed intersections by the path through the city centre may be greater than those passed on the by-pass roads, the total travel time of the former path may be greater than the latter path. In that case, the Shortest Time Path (STP) may be the LNP instead of the SDP. For example, in Figure 1, if one travels from  $s$  to  $t$  through the city centre, the total number of intersections crossed along the straight line connecting  $s$  and  $t$  is six. Obviously, this is greater than the total passed intersections along the upper and lower circle by-pass roads. Therefore, the LNP gives comparative information for evaluation against the shortest distance path to assist drivers in selecting driving routes.

The LNP problem can be seen as finding the SP in a network where all attributes associated with links are the same or are not associated with any attributes at all. Using Figure 2 as an example, the LNP problem can be defined as finding the shortest path in the network depicted in Figure 2 where all links have no associated attribute.



Based on how many SPs need to be solved, the SP problem can be classified into three types, namely *one-to-one*, *one-to-many*, and *all-to-all* SP problems. The one-to-one SP problem refers to the cases where there is only one origin node and one destination node. The SOPO and MOPO problems can be categorised as the one-to-one SP problem. There are two major approaches for solving the one-to-one SP problem, namely the *label-setting approach* and the *label-correcting approach*. The former approach is best represented by Dijkstra's (1959) algorithm, which can be applied to networks with non-negative-length links. The latter approach, represented by Bellman's (1958) and Moore's (1957) algorithms may be used in networks with negative-length and non-negative-length links. However, neither approach is applicable for networks with loops or a negative cycle.

The attributes associated with links are the keys to determining the SPs. However, when all attributes have the same value, say 1, neither label-setting and label-correcting algorithms may be applicable for solving the SP problem. This is because all the cumulative distances of all the links connecting to each scan node are the same. There is no additional information to assist the SP algorithm in choosing the current shortest distance. Therefore, SP algorithms are no longer useful for solving the LNP problem. To solve the LNP problem, in this paper, an algorithm based on *set operation* is proposed. The set operation first scans from the start node in *breadth-first-search* order and uses two lists: one to record all scanned nodes and another for the nodes that are going to be scanned in the next step. Then the scan process is repeated until the destination node is reached. Once the destination node is reached, the LNP can be obtained by sorting the set of scanned nodes in connecting order. The algorithm for solving the LNP problem is as follows:

$G$ : a network with  $n$  nodes

$S$ : start node of a path

$T$ : destination node of a path

$G^\alpha$ : a set of nodes with  $i$  number of members currently being traced

$G^\beta$ : all nodes connecting with  $G^\alpha$

$G_i^\beta$ : nodes connecting with the  $i$ th  $G^\alpha$

$G^\lambda$ : nodes which have been traced

$G_j^\lambda$ : nodes which have been traced in the  $j$ th run

Initialisation:

$G^\alpha \leftarrow S$

$G^\beta \leftarrow \emptyset$

$G^\lambda \leftarrow \emptyset$

$j=0$

Main loop:

DO WHILE  $G^\alpha \text{ NOT } \emptyset$

$j=j+1$

$i = \text{member of } G^\alpha$

FOR 1 TO  $i$

$G_i^\beta \leftarrow \text{get nodes connecting with } i\text{th } G^\alpha \text{ and } \notin G^\lambda$

$G^\beta \leftarrow \{G_i^\beta \cup G^\beta\}$

IF  $T \in G_i^\beta$  THEN

report the path with least nodes by sorting  $G^\alpha$

```

EXIT DO
END IF
END FOR
 $G_j^\lambda \leftarrow G^\alpha$ 
 $G^\lambda \leftarrow \{G^\alpha \cup G^\lambda\}$ 
 $G^\alpha \leftarrow \{G^\alpha \cup G^\beta\} - \{G^\alpha \cap G^\beta\} \notin G^\alpha$ 
END DO

```

### 3.3. Solving the Minimum-Turn Path Problem.

3.3.1. *Measuring the Turns.* As mentioned, a *turn* can be represented with two links or three nodes. However, before one can use nodes and links to represent a turn, one should determine the definition of a turn in a real road network. In principle a turn can be identified and measured in two ways, *manually* by human knowledge or *automatically* by measuring the angle change of driving direction. The former approach can be done by humans scanning the whole real road map to retrieve turns and then recording this information in the network topology with '1' representing the situation where there is a turn between two links and '0' where there is no turn. The latter approach can be implemented by calculating the angle between two links. However, it is difficult to identify a value for which an angle change can be classified as a turn.

Take both *cross* and *Y-type road junctions* for example, as depicted in Figures 3 and 4 respectively. Obviously, there are eight valid turns in Figure 3 including  $\langle a \rightarrow b' \rangle$ ,  $\langle a \rightarrow c \rangle$ ,  $\langle b \rightarrow d \rangle$ ,  $\langle b \rightarrow a' \rangle$ ,  $\langle d' \rightarrow c \rangle$ ,  $\langle d' \rightarrow b' \rangle$ ,  $\langle c' \rightarrow a' \rangle$  and  $\langle c' \rightarrow d \rangle$ . However, in the case of Figure 4, one may recognise that there are six valid turns, i.e.  $\langle a \rightarrow b \rangle$ ,  $\langle a \rightarrow c' \rangle$ ,  $\langle b' \rightarrow a' \rangle$ ,  $\langle b' \rightarrow c' \rangle$ ,  $\langle c \rightarrow a' \rangle$  and  $\langle c \rightarrow b \rangle$ . But it may also be argued that turn  $\langle c \rightarrow a' \rangle$  should be excluded depending on the acceptable threshold of the *angle-change* to be utilised in identifying a turn. Since a path is composed of many links, it is possible to measure the total turn angle by calculating the cumulative angle change of the driving direction. Furthermore, by comparison with a turn definition using a 0–1 integer value retrieved manually in advance, it is a better approach to transform the MTP problem into the *minimum turn-angle problem*, thus minimising the cumulative angle-change of a path.

In order to use the cumulative change of driving direction as the measure for the minimum-turn objective, the angle of each driving direction has to be determined before the SP algorithm can be applied. Using Figure 3 as an example, each driving direction can be represented as a vector in the format from node to node. Suppose that the points *A* and *C* in Figure 2 have coordinates  $(x_1, y_1)$  and  $(x_2, y_2)$ , respectively. Then the direction  $\theta_{ac}$  of vector *AC* can be represented by:

$$\theta_{ac} = \cos^{-1}(x_2 - x_1)/r \quad (3.1)$$

Note that the parameter *r* represents the Euclidean distance from point *A* to point *C* and is the Pythagorean distance, given by:

$$r = \sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]} \quad (3.2)$$

Assume that the network in Figure 2 is symmetrical. Then each link is associated with two vectors that represent corresponding driving directions. The whole network and its vectors can be depicted as in Figure 5.

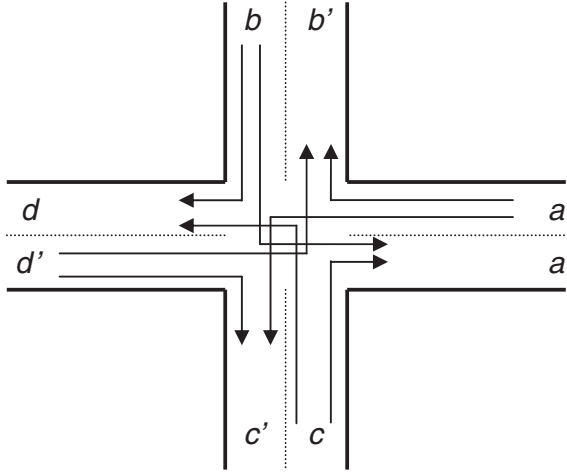


Figure 3. An example of turns in a cross road junction.

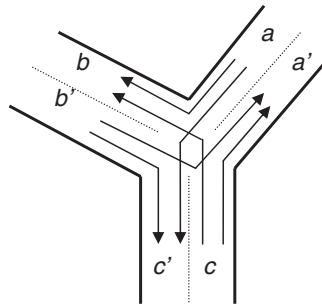


Figure 4. An example of turns in a Y-type road junction.

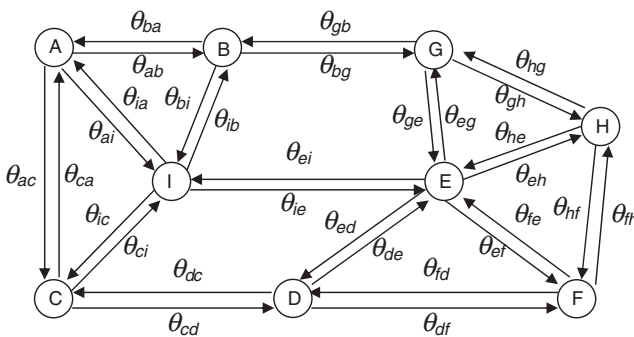


Figure 5. Vectors representing driving directions.

3.3.2. *Algorithm for Solving the MTP Problem.* All SP algorithms for solving the one-to-one SP problem initialise a zero value for the start node and label the node currently with the shortest distance. Unlike the SP problem, there are at least three links in a start node for the MTP problem, each of which is a possible driving direction

with a possible non-zero initialisation value. Therefore, to solve the MTP problem each driving direction of the start node has to be considered as having the same role as the start node in the SP problem. In other words, if one applies an SP algorithm to solve a MTP problem, the MTP problem should first be divided into at least three *sub-problems*. Then the final solution can be obtained from the sub-problem which has the minimum value. Taking the path from start node  $A$  to destination node  $F$  in Figure 5 as an example, there are three driving directions denoted by vectors  $AB$ ,  $AC$  and  $AI$ . The MTP problem can be divided into three sub-problems, namely finding the MTPs of vector  $AB$  to node  $F$ , vector  $AC$  to node  $F$ , and vector  $AI$  to node  $F$ . To obtain the solution of the MTP problem, SP problems vector  $AB$  to node  $F$ , vector  $AC$  to node  $F$ , and vector  $AI$  to node  $F$  should be solved.

To solve each sub-problem, first, the start node is initialised with one of its connecting vectors. Taking the third sub-problem as an example, the initialisation value of node  $A$  is the  $\theta_{ai}$  radius. Second, set node  $I$  as the new start node and label it. Then use the  $\theta_{ai}$  radius as the origin direction to determine which vectors have the smallest angle difference. Label the node connecting to the new start node with the smallest angle difference. Repeat the second step until the destination node is reached. For the first sub-problem it is obvious that the MTP is composed of nodes  $A$ ,  $B$ ,  $G$ ,  $H$  and  $F$ . For the second sub-problem, clearly the MTP is concerned with nodes  $A$ ,  $C$ ,  $D$ , and  $F$ . However, the third sub-problem becomes ambiguous when traversing to node  $E$ , if the angle difference of vectors  $EH$  and  $EF$  are the same. To cope with such a problem, a directing vector defined by node  $E$  to the destination node (in this case, node  $F$ ) can be used to determine which vector should be chosen. The directing procedure is achieved by using both vectors  $IE$  and  $IF$  to determine the vector with the smallest angle difference.

4. EXPERIMENTAL ANALYSIS. Clearly, as the number of nodes and links in a road network increases, the total number of decision variables in the MOPO and SOPO problems will also increase. As the number of decision variables increases, solutions to the MOPO and SOPO problems can become intractable. Manual solutions may only be feasible in cases which involve a small number of nodes and links. Solutions to problems involving large road networks therefore require the assistance of GIS software. A program which integrated MapInfo GIS software for solving the MOPO and SOPO problems was developed in order to facilitate empirical studies using real road networks. In addition, because the decision variables of the objectives of cumulative distance, roadway grade and roadway capacity are based on links and share the same problem solving methods, in order to reduce computational load during experimental analysis, only cumulative distance was used. Hence, only three of the five objectives of the proposed model were selected for the empirical studies. They relate to the SDP, LNP and MTP problems. All experiments were carried out on the test bed with an Intel Xeon 3.2 GHz CPU and 2.0 GB of RAM running Microsoft Windows XP Professional.

4.1. *Experimental Road Networks*. Two sub-networks of the metropolitan area of Taichung, located in central Taiwan, covering an area of approximately 2,221.25 square km and with a population of 2.5 million, were selected to conduct the experiments. The entire road network of metropolitan Taichung and the two sub-networks indicated by the two boxes are illustrated in Figure 6. The first sub-network

Table 1. Node and link number and connecting-ratio of the experimental road networks.

Road Network Types	Total Nodes	Total Links	Connecting-Ratio
entire road network	27,641	36,499	2.64
radial-circumferential road network	1,115	1,447	2.01
grid-type road network	693	1,037	2.95



Figure 6. Empirical test road networks.

(top box), referred to as the radial-circumferential road network, as illustrated in Figure 7, is the largest suburb of metropolitan Taichung, with a total population of about 150 thousand people. The second sub-network (lower box), referred to as the grid-type road network, illustrated in Figure 9, is located in the downtown area of Taichung. The total number of nodes, links and average links connected by a node (the *connecting-ratio*) for the entire road network and two sub-networks are listed in Table 1.



Figure 7. The optimal paths of SDP, LNP, MTP and MOPO problems in a radial-circumferential road network.

4.2. *Experimental Study of the SOPO Problems.* Two links from the radial-circumferential road network were selected (illustrated in Figure 7) and resulted in the identification of optimal paths for the SDP, LNP and MTP problems. The results are illustrated in Figures 7 and 8. Note that the optimal paths for these three problems are each displayed in the MapInfo map window (Figure 7) and in the message window (Figure 8). The former shows the intersections of the optimal paths in the experimental road network, while the latter displays each optimal path's detailed nodes, objective values and run time. The two thicker lines were used to indicate the pair  $(s, t)$ ; the IDs of nodes and the objective values for each optimal path are displayed in the MapInfo message window as a serial list separated by semicolons; except for the nodes' list, the first item of all lists represent the sum of objective values. The units for measuring each objective's value of the SDP, LNP and MTP problems are metre, node and degree ( $360^\circ$ ) respectively. The run time displayed in the message window is the elapsed time of the entire calculation, including: the time for loading the network topology to memory; the time for identifying the location of the pair  $(s, t)$ ; the time for searching the optimal paths of the SDP, LNP and MTP problems; and the time for converting the optimal paths from the network topology format to MapInfo's file format.

As mentioned, if the pair  $(s, t)$  is located at both sides of a radial-circumferential road network the SDP might go through the city centre along the straight line connecting  $s$  and  $t$ . The LNP, however, may utilise the by-pass roads. Therefore, the total passed intersections by the SDP may be greater than those passed by the LNP,



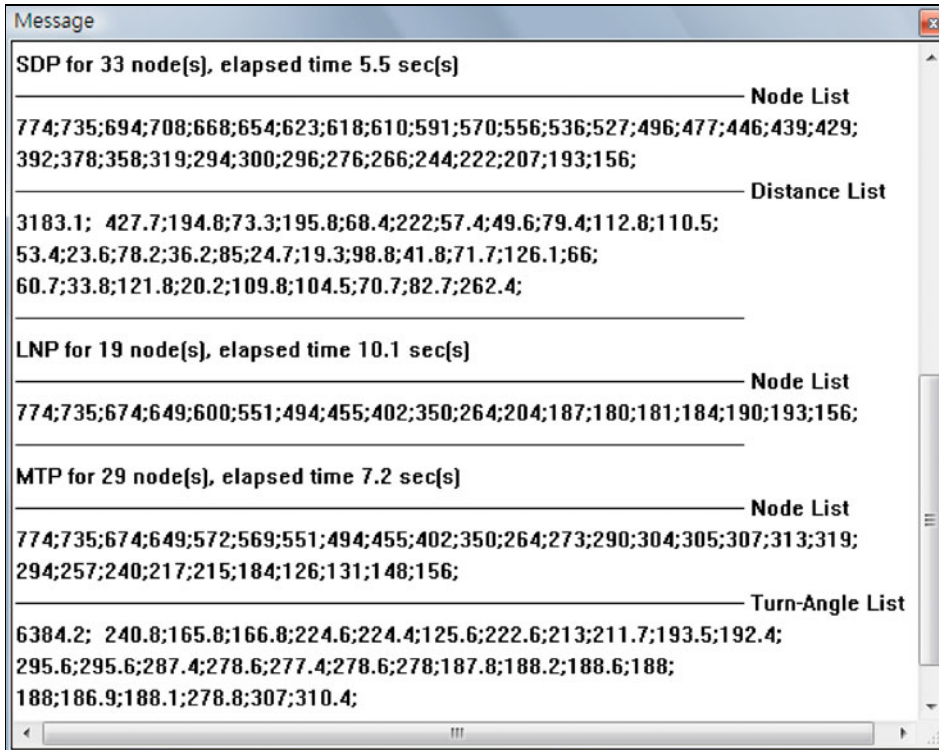


Figure 8. List of nodes and each objective’s values of SDP, LNP and MTP in a radial-circumferential road network.

and the cumulative distance of the LNP may be greater than for the SDP. According to the results of Figures 7 and 8, it can be seen that the SDP and LNP do result in different paths in a radial-circumferential road network. Furthermore the total number of passed intersections by the SDP (33 nodes) is greater than those passed by the LNP (19 nodes). Despite the many traffic signals the total travel time of the SDP may be greater than for the LNP, since the total number of intersections of the SDP is greater than for the LNP. In that case, the shortest-time path may be the LNP instead of the SDP. The LNP provides comparative information for evaluation against the SDP to assist drivers in selecting their driving routes. This example illustrates the above-mentioned scenario. As mentioned, due to the difficulty in identifying a specific number of degrees at which an angle change can be classified as a turn, this study uses the cumulative angle change of the driving direction to measure the minimum-turn objective. Using the same pair (*s*, *t*), the MTP is also illustrated in Figures 7 and 8. A turn angle is calculated by taking the difference of two adjacent items of the MTP in Figure 8. The unambiguous turns (i.e. the turn angle is close to 90° or 180°), which may be clearly identified, are six in number and are denoted by the solid black circles in Figure 7. However, a turn angle of about 57.8° (166.8; 224.6), indicated by the asterisk, is ambiguous and difficult to classify. This example highlights the difficulty mentioned earlier, and confirms that the cumulative angle change of the driving direction is a better approach for resolving the MTP objective.

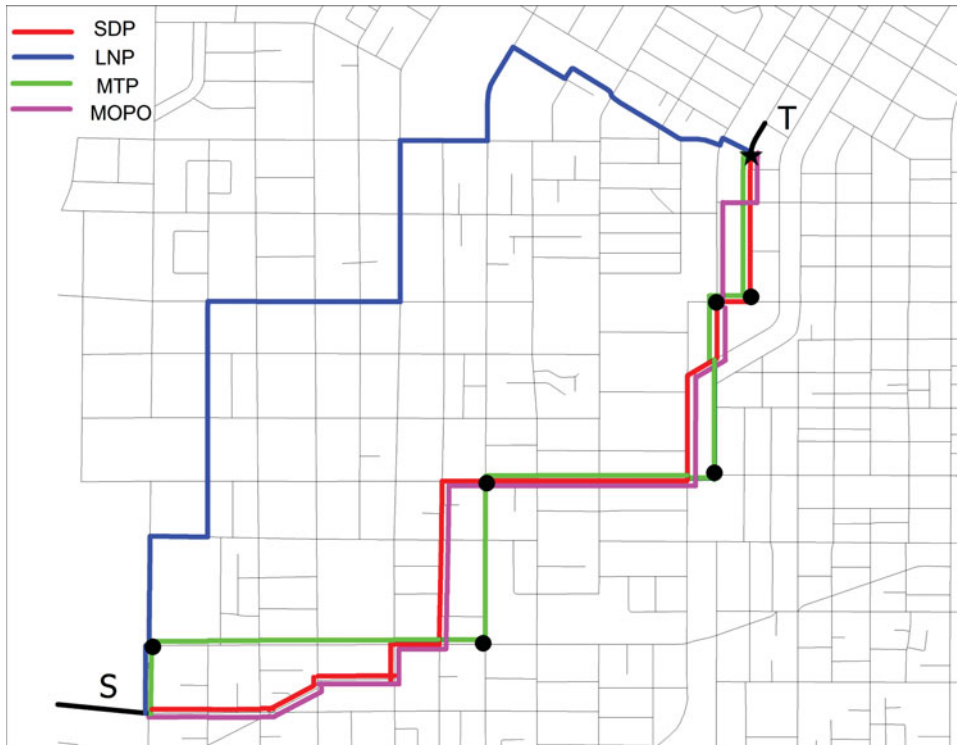


Figure 9. The optimal paths for SDP, LNP, MTP and MOPO problems in a grid-type road network.

The same experiments were therefore conducted using a grid-type road network. The results are illustrated in Figures 9 and 10. As in the case of the radial-circumferential road network, the results also reveal that the SDP and LNP do appear on different paths in a grid-type road network; the total number of passed intersections by the SDP (36 nodes) is greater than those passed by the LNP (27 nodes). The unambiguous turns, which may be clearly identified, are six in number and are denoted by the solid black circles in Figure 9. However, a turn angle, about  $23.5^\circ$  ( $88.1; 64.6$ ), indicated by the asterisk is ambiguous.

4.3. *Experimental Study of the MOPO Problem.* Unlike solving SOPO problems where the pair  $(s, t)$  is the only parameter that needs to be input, to solve the MOPO problem more parameters have to be assigned, such as the weighting values for each SOPO objective. It is important to be aware of the fact that the weighted-sum method is generally used to approximate the non-inferior set by parametrically varying the weights  $w_i$  with a predetermined step value between one to zero if there are more than two decision-makers involved in the decision-making process. However, this is not an efficient method for seeking an exact representation of the non-inferior set since a number of different sets of weights need to be tried until an adequate representation of the non-inferior set is obtained. This means that as higher degrees of accuracy are requested in the approximation, the computational burden will increase. On the other hand, if the weights are obtained directly from decision-makers, the above

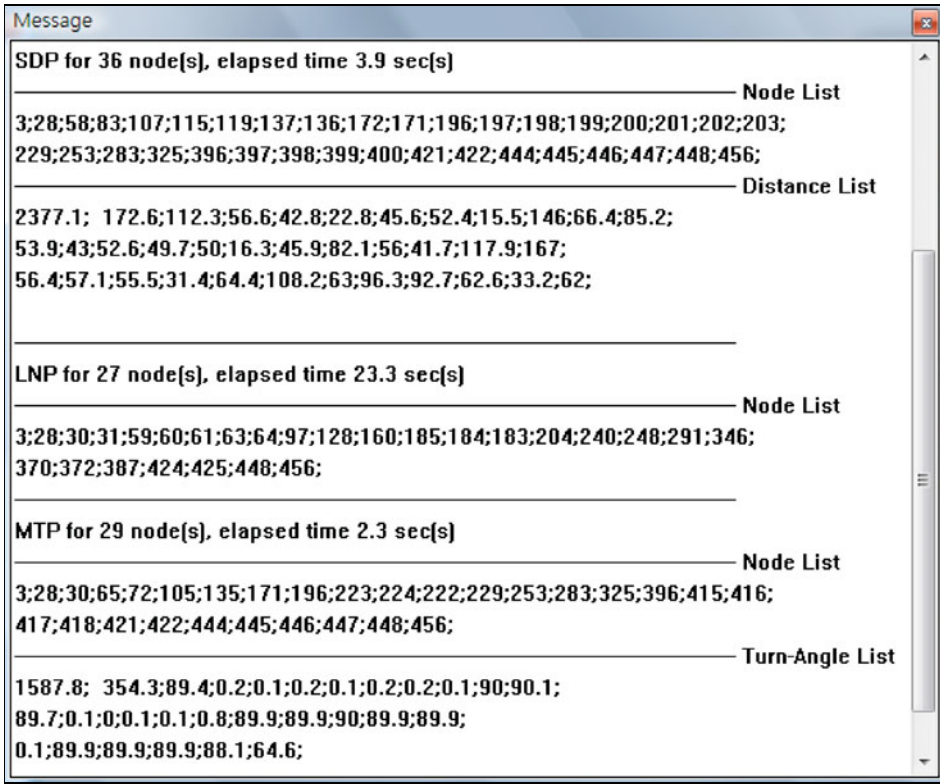


Figure 10. List of nodes and each objective's values of SDP, LNP and MTP in a grid-type road network.

disadvantages will be overcome. Since the proposed MOPO decision will only involve one decision-maker, the weighting values can be obtained directly from the driver.

Experiments for solving the MOPO problem were conducted using both the radial-circumferential and grid-type road networks. In addition, the same links as those used for the experiments on SOPO problems were used. As before, the results for the MOPO problem is displayed in two ways: in the MapInfo map window (Figures 7 and 9), and the message window (Figures 11 and 12). Note that the weighting values of each objective are set to 0.5, 0.25 and 0.25 for the SDP, LNP and MTP objectives respectively. To make comparison easier, the optimal paths of the MOPO problem and the optimal paths of all SOPO problems are displayed together in Figures 7 and 9 and the MOPO's objective values were converted to each objective's corresponding value and plotted in Figures 11 and 12. Since the optimal path of the MOPO problem is a compromise solution, the converted objective values will all be greater than or equal to the corresponding optimal solutions of the SOPO problems. According to the experimental results, the converted objective values for each objective of the MOPO are *distance* = 3183.14, *node* = 33, and *turn angle* = 6826.5; and *distance* = 2377.79, *node* = 36 and *turn angle* = 2479.2 for the radial-circumferential and grid-type road networks respectively. Note that these values are greater than or equal to the

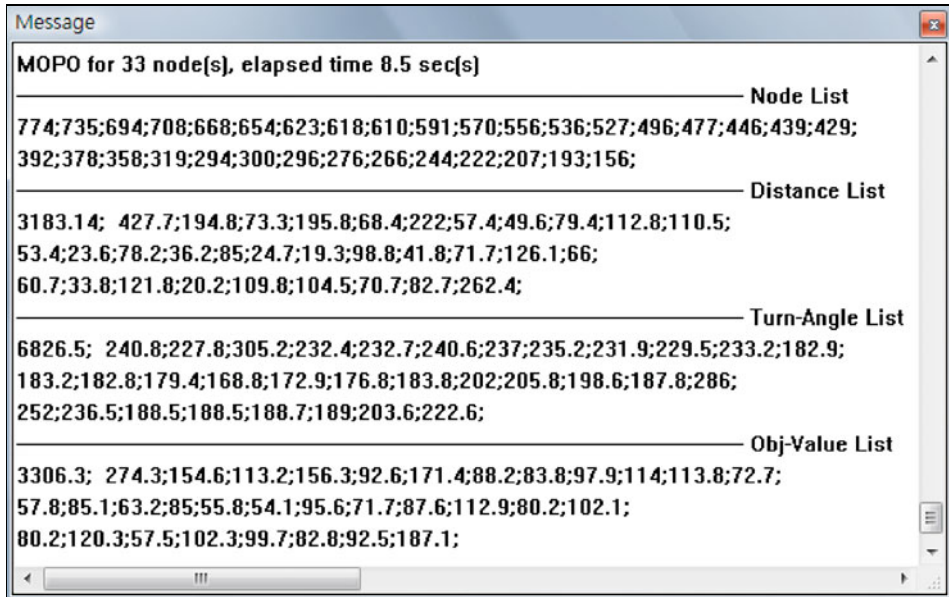


Figure 11. List of nodes and objectives' values of MOPO path in a radial-circumferential road network.

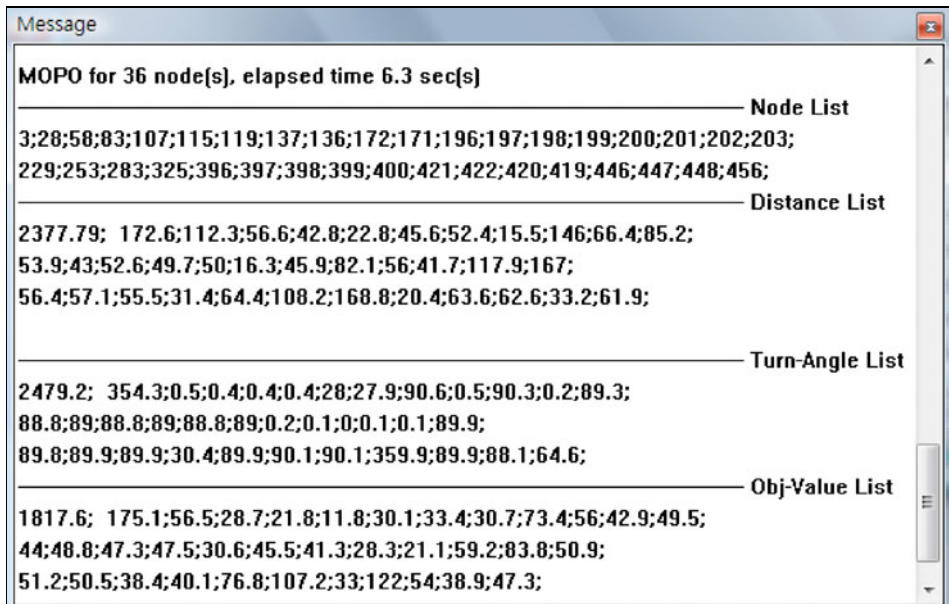


Figure 12. List of nodes and objectives' values of MOPO path in a grid-type road network.

corresponding optimal values of the SOPO problems that are  $distance = 3183.14$ ,  $node = 19$ , and  $turn\ angle = 6384.2$ ; and  $distance = 2377.79$ ,  $node = 27$  and  $turn\ angle = 1587.8$  for the radial-circumferential and grid-type road networks respectively.

The above experiments demonstrate how the proposed MOPO and SOPO models could be implemented within a GIS as a specialised spatial analysis tool to generate optimal solutions. They also demonstrate the advantages of integration with the GIS in supporting both spatial and attribute data displays. Contemporary commercial car navigation systems typically only provide drivers with the shortest-distance guiding function. The proposed MOPO and SOPO models not only give drivers more diverse selection criteria for their driving routes, but also provide transportation planners with a useful approach to solving the traffic assignment problems in a way which more closely simulates real driver decision behaviour.

5. CONCLUSION. Car navigation systems and tourist information services have been recognised as two of the most useful applications of *location-based services* (Jeong et al., 2006). One of the most common pieces of information that drivers and tourists usually request is route direction. The key technologies for supporting this request are the Shortest Path (SP) algorithms. Currently the only Shortest Distance Path (SDP) is provided by commercial products where the current SP algorithms only support the single-objective decision model. To extend the current single-objective model to a multi objective decision model, this paper proposed a Multi Objective Path Optimisation (MOPO) model and supplementary Single Objective Path Optimisation (SOPO) models. Appropriate Multi Objective Programming (MOP) techniques for solving the MOPO model were discussed and suggested. Algorithms for solving the Least Node Path (LNP) and Minimum Turn Path (MTP) problems were presented. To demonstrate the applicability of the proposed MOPO and SOPO models, MOP techniques and SDP, LNP and MTP algorithms, experiments were conducted.

Through these theoretical and empirical studies several useful conclusions were drawn. Firstly, the experimental results demonstrate the advantages of integration with a commercial Geographic Information System (GIS) package in supporting both spatial and attribute data displays. It can be claimed that by obtaining assistance from the GIS software it is easy for drivers to obtain the optimal paths of the SDP, LNP, MTP and MOPO problems very quickly, despite these problems being highly complex and difficult to resolve manually. Secondly, according to the experimental results, the proposed LNP, MTP and MOPO decision models provide drivers richer information, enabling them to choose their driving routes in more diverse ways. Finally, it is shown by the experimental results that the SDP and LNP mostly locate on different paths in both radial-circumferential and grid-type road networks, and the total passed intersections by the SDP are greater than passing by the LNP.

## REFERENCES

- Bellman, R. (1958). On a routing problem, *Quarter Application Mathematic*, **16**, 87–90.
- Benayoun, R., Montgolfier, J., Tergny, J., and Laritchev, O. (1971). Linear programming with multiple objective functions: step method (STEM), *Mathematical Programming*, **1**, 366–375.
- Ben-Tal, A. (1980). Characterization of pareto and lexicographic optimal solutions, in Fandel, G, and Gal, T, editors, *Multiple Criteria Decision Making Theory and Application*, volume 177 of *Lecture Notes in Economics and Mathematical Systems*, 1–11, Springer-Verlag, Berlin.
- Berger, J. O. (1980). *Statistical decision theory: foundations, concepts and methods*, Springer-Verlag, New York.
- Brownlow, S. A., and Watson, S. R. (1987). Structuring multi-attribute value hierarchies, *Journal of the Operational Research Society*, **38**, 309–317.



- Chakraborty, B., Maeda, T., and Chakraborty, G. (2005). Multiobjective route selection for car navigation system using genetic algorithm, *Proceedings of the 2005 IEEE mid-summer workshop on soft computing in industrial applications*, Espoo, Finland.
- Charnes, A., Cooper, W. W., Niehaus, R. J., and Stedry, A. (1969). Static and dynamic assignment models with multiple objectives and some remarks on organization design, *Management Science*, **15**, B365–B375.
- Climaco, J., and Martins, E. (1982). A bicriterion shortest path algorithm, *European Journal of Operation Research*, **11**, 399–404.
- Cohon, J. L. (1978). *Multiobjective programming and planning*, Academic Press, New York.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs, *Numerische Mathematil*, **1**, 269–271.
- Goicoechea, A., Duckstein, L., and Fogel, M. (1979). Multiple objective under uncertainty: an illustrative application of PROTRADE, *Water Resources Research*, **15**, 203–210.
- Goicoechea, A., Hansen, D. R., and Duckstein, L. (1982). *Multiobjective analysis with engineering and business applications*, John Wiley and Sons, New York.
- Hallam, C., Harrison, K. J., and Ward, J. A. (2001). A multiobjective optimal path algorithm, *Digital Signal Processing*, **11**, 133–143.
- Haimes, Y. (1973). Integrated system identification and optimization, in Leonides, C editor, *Control and Dynamic Systems: Advances in Theory and Applications*, 9, 435–518, Academic Press, New York.
- Hansen, P. (1980). Bicriterion path problems, in Fandel, G, and Gal, T, editors, *Multiple Criteria Decision Making Theory and Applications*, Proceedings 1979, 109–127.
- Henig, M. I. (1985). The shortest path problem with two objective functions, *European Journal of Operational Research*, **25**, 281–291.
- Hwang, C. L., and Lin, M. J. (1987). *Group decision-making under multiple criteria*, Springer, New York.
- Jeong, C. W., Chung, Y. J., Joo, S. C., and Lee, J. H. (2006). Tourism guided information system for location-based services, *Lecture Note in Computer Science*, **3842**, 749–755.
- Keeney, R. L., and Raiffa, H. (1976). *Decisions with multiple objectives: preferences and value trade-offs*, John Wiley and Sons, New York.
- Monarchi, D. E., Kisiel, C. C., and Duckstein, L. (1973). Interactive multiobjective programming in water resources: a case study, *Water Resources Research*, **9**, 837–850.
- Moore, E. F. (1957). The shortest through a maze, in *Proceeding of the International Symposium on the Theory of Switching*, 285–292.
- Rao, S. S. (1996). *Engineering optimization*, John Wiley and Sons, New York.
- Salukvadze, M. E. (1974). On the existence of solutions in problems of optimization under vector value criteria, *Journal of Optimization Theory and Applications*, **12**, 203–217.
- Sarma, G. V., Sellami, L., and Houam, K. D. (1993). Application of lexicographic goal programming in production planning – two case studies, *Operational Research*, **30**, 141–162.
- Zadeh, L. A. (1963). Optimality and non-scalar-valued performance criteria, *IEEE Transactions on Automatic Control*. **AC-8**, 59–60.