Improved solutions for the Chinese postman problem on mixed networks

W.L. Pearn[a,*], J.B. Chou[b]

[a]*Department of Industrial Engineering & Management, National Chiao Tung University, Hsinchu, Taiwan ROC*
[b]*Manufacturing & Material Division, Actima Technology Corporation, Hsinchu, Taiwan ROC*

Abstract

The Chinese postman problem (CPP) is that of finding the shortest postman tour covering all the roads in the network. The CPP is polynomial-time solvable on totally directed or undirected networks, but is NP-complete on mixed networks (MCPP). Several heuristic solution procedures including Mixed-1, Mixed-2, Modified Mixed-1, and Modified Mixed-2 algorithms, have been proposed to solve the problem approximately. In this paper, we briefly review these four existing algorithms, then propose two improvement procedures to improve the solutions. The proposed procedures are tested and compared with the four existing algorithms. The results show that the proposed procedures significantly improve the existing solutions. © 1999 Elsevier Science Ltd. All rights reserved.

Scope and purpose

Given a road network, the Chinese postman problem (CPP) is to find the shortest postman tour covering all the roads in the network. The CPP is polynomial-time solvable on directed or undirected networks, but is NP-complete on mixed networks (MCPP). On mixed networks, some roads are allowed to be traversed in both directions, and others may be traversed in one specified direction only. Applications of the MCPP include routing of mail-delivery vehicles, parking-meter coin collection or household refuse collection vehicles, street sweepers, snow plows, school buses; inspection of electric power or oil lines, and reading electric meters. The purpose of this paper is to review the existing solution procedures, and introduce two new procedures to improve the existing solutions.

*Corresponding author. Tel.: 00886 35 714261; fax: 00886 35 614387; e-mail:roller@cc.nctu.edu.tw.

1. Introduction

Given a road network, the Chinese postman problem (CPP) is that of finding the shortest postman tour covering all the roads in the network. The CPP on totally directed or undirected networks can be solved optimally using the efficient matching algorithm developed by Edmonds and Johnson [1]. But, the CPP on mixed networks (MCPP) is NP-complete [2]. On mixed networks, some roads (called edges) are allowed to be traversed in both directions with equal distance, and others (called arcs) may be traversed in one specified direction only. Such a problem, of course, reflects real-life situations more directly. In particular, for cities with narrow streets and only one-way traffic can be designed, or where there is a need for traffic-flow control. Real-world applications of the MCPP include routing of mail-delivery vehicles, household refuse collection vehicles, street sweepers, school buses; spraying roads with salt, inspection of electric power lines, and many others.

The MCPP may be briefly defined as follows. We are given a road network $G = (V, E, A)$ with V representing the set of nodes, E representing the set of edges, and A representing the set of arcs. For each edge $(i, j) \in E$ we have distances $d(i, j)$ and $d(j, i)$ with $d(j, i) = d(i, j)$. For each arc $(i, j) \in A$, we have a distance $d(i, j)$ from i to j, with $d(j, i)$ *setting to infinity in this case*. Then, the MCPP is to find a postman tour starting from the depot, traversing the edges and arcs in $E \cup A$ at least once, and returning to the depot with total distance traversed minimized.

Kappauf and Koehler [3] formulated the MCPP as an integer linear programming, and established a one-to-one correspondence between the extreme points of linear programming polyhedron and prime assigned Euler networks. Ralphs [4] showed that the extreme points of the linear relaxation polyhedron of the integer linear programming are all half-integral. Christofides et al. [5] developed an exact algorithm to solve the MCPP optimally. The algorithm is essentially based on a branch-and-bound algorithm using Lagrangean relaxations. Minieka [6] presented a transformation converting the MCPP into a flow-with-gains problem. The conversion allows the MCPP to be solved optimally using available linear programming and cutting-plane techniques. Nobert and Picard [7] also developed an LP-based cutting-plane algorithm to solve the MCPP optimally. Unfortunately, these methods are computationally inefficient; only problems of small or moderate size may be solved optimally. Extensions of the MCPP (such as the mixed rural postman problem) are considered by Anily et al. [8]. Because of the problem's complexity, heuristic algorithms including Mixed-1, Mixed-2, modified Mixed-1, and modified Mixed-2, have been proposed to solve the MCPP approximately (see [1, 5, 9–11]). In this paper, we present two improvement procedures to improve the existing solutions.

2. The Mixed-1 algorithms

Edmonds and Johnson [1] presented a solution procedure (called Mixed-1) to solve the MCPP approximately. Since the network for the generated Mixed-1 solution may not be even, Frederickson [9] adds a new phase to Mixed-1 algorithm which recovers the resulting network back to even. We note that the sufficient and necessary condition for the existence of a postman route on mixed networks is that the network must be even and symmetric.

2.1. Mixed-1 algorithm (1A)

Mixed-1 algorithm consists of three phases. Phase I converts the original network into an even one by treating all the directed arcs as undirected edges, then solves the CPP on the even network. Phase II transforms the CPP solution into a symmetric network (may not be maintained even). Phase III recovers the transformed symmetric network back to even so that a postman tour may be constructed. The performance of Mixed-1 algorithm, in the worst case, is bounded by two. That is, (Mixed-1 solution)/(optimal solution) $\leqslant 2$, and the bound is approachable [9].

2.2. Modified Mixed-1 (1B)

Pearn and Liu [11] observed that the artificial edges/arcs generated from the matching and minimal-cost flow sections may form artificial cycles. Thus, a modification by removing the artificial cycles is proposed to improve the solution. Since only redundant artificial cycles are removed, we then have (modified Mixed-1 solution) $\leqslant$ (Mixed-1 solution).

2.3. Improved Mixed-1 (1C)

To further improve the solution, we propose the following modification. Let E_A be the artificial arcs added to G forming the modified Mixed-1 solution G_3. Perform the following procedure: (1) find the minimum-cost flow solution E_F over the network $G_3 - E_A$ (which is totally directed) obtained from G_3 by removing the artificial arcs E_A; (2) replace E_A by E_F to obtain the new solution $(G_3 - E_A) \cup E_F$. For edges having two different directions in G_3, each direction will be chosen to generate a complete solution, and the best of the two is selected. Since all artificial arcs (E_A) in the modified Mixed-1 solution are removed and replaced by the minimal-cost flow solution (E_F), (improved Mixed-1 solution) $\leqslant$ (modified Mixed-1 solution).

Example 1. Consider the network depicted in Fig. 1a with depot node 1 and edge (arc) lengths $d(1, 2) = d(3, 4) = d(4, 3) = 1$, $d(1, 3) = d(2, 3) = d(3, 2) = d(2, 4) = d(4, 2) = 2$, and $d(4, 1) = M$. It is straightforward to verify that Mixed-1 solution (Fig. 1b) is $3M + 10$, the modified Mixed-1 solution (Fig. 1c) is $3M + 10$, and the improved Mixed-1 solution (Fig. 1d) is $2M + 10$ which is also optimal. The improved Mixed-1 significantly outperforms Mixed-1 and the modified Mixed-1 in this example. In fact, for sufficiently large value of M, (improved Mixed-1 solution)/(Mixed-1 solution) = (improved Mixed-1 solution)/(modified Mixed-1 solution) $\cong 2/3$.

Example 2. Consider the network depicted in Fig. 2a with depot node 1 and arc (edge) lengths $d(3, 2) = d(5, 4) = 1$, $d(2, 4) = d(4, 2) = d(3, 5) = d(5, 3) = 2$, and $d(1, 2) = d(2, 1) = d(1, 3) = d(3, 1) = d(4, 6) = d(6, 4) = d(5, 6) = d(6, 5) = M$. It is easy to verify that Mixed-1 solution (Fig. 2b) is $8M + 8$, the modified Mixed-1 solution (Fig. 2c) is $6M + 7$, and the improved Mixed-1 solution (Fig. 2d) is $4M + 10$ which is also optimal. We note that the improved Mixed-1 significantly outperform Mixed-1 and the modified Mixed-1 again. For sufficiently large M, (improved Mixed-1 solution)/(Mixed-1 solution) $\cong 1/2$, and (improved Mixed-1 solution)/(modified Mixed-1 solution) $\cong 2/3$.
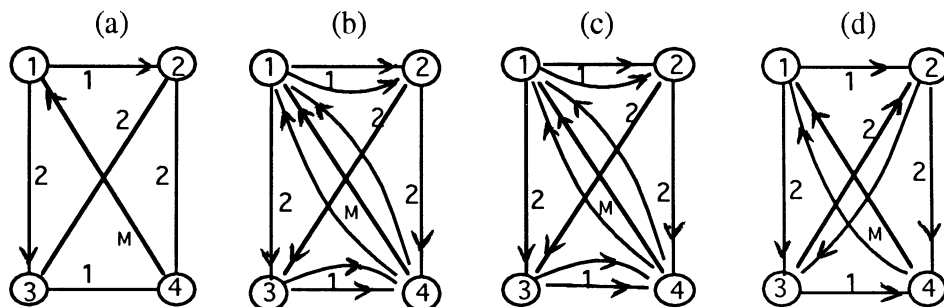
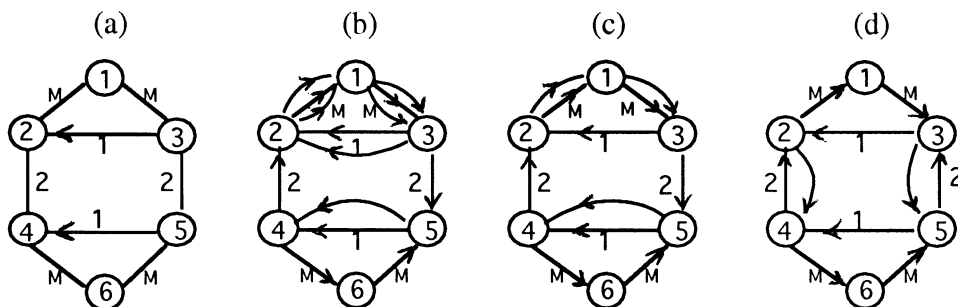Fig. 1a–d. Example 1 and the three Mixed-1 solutions.

Fig. 2a–d. Example 2 and the three Mixed-1 solutions.

To compare the improved Mixed-1 with Mixed-1 and the modified Mixed-1, we run the 60 problems which have been tested by Pearn and Liu [11]. We also extend the testing by expanding the number of problems to 120. The sizes of the problems range from 10 to 35 nodes ($10 \leqslant |V| \leqslant 35$), 20 to 380 edges and arcs ($20 \leqslant |E| + |A| \leqslant 380$). These problems are all randomly generated, including 40 problems of $70\% < P \leqslant 100\%$ (category $P1$), 40 problems of $40\% < P \leqslant 70\%$ (category $P2$), and 40 problems of $0\% < P \leqslant 40\%$ (category $P3$), where $P = |A|/(|A| + |E|)$, $|A|$ is the number of arcs, and $|E|$ is the number of edges. Table 1 displays the performance comparisons among the three Mixed-1 solutions in terms of (1) average percentage above the lower bound, (2) worst percentage above the lower bound, (3) number of problems receiving the best solutions, and (4) number of problems achieving the problem lower bounds (hence the solutions must be optimal). The results indicate that:

(a) the improved Mixed-1 improves Mixed-1 by 8.3%, 2.7%, 0.6%, and improves the modified Mixed-1 by 3.5%, 1.4%, 0.6% on problems in category $P1$, category $P2$, and category $P3$, respectively in terms of average percentage above the lower bound.
(b) the improved Mixed-1 reduces 'the worst percentage above the lower bound' by 20.7%, and 12.8% on problems in category $P1$, and category $P3$, respectively.
(c) the improved Mixed-1 improves the modified Mixed-1 for 38, 24, and 2 problems on problems in category $P1$, category $P2$, and category $P3$, respectively, in terms of number of best solutions.

Table 1
Comparisons among the three Mixed-1 solutions

Problems		1A	1B	1C
P1	Average % above the lower bound	34.20	29.42	25.90
	Worst % above the lower bound	97.33	83.33	62.67
	Number of best solutions	0	2	40
	Number of optimal solutions	0	0	0
P2	Average % above the lower bound	7.60	6.32	4.95
	Worst % above the lower bound	63.64	42.73	42.73
	Number of best solutions	12	16	40
	Number of optimal solutions	6	6	6
P3	Average % above the lower bound	0.75	0.75	0.13
	Worst % above the lower bound	16.04	16.04	3.21
	Number of best solutions	38	38	40
	Number of optimal solutions	38	38	38

Table 2
Improvement made by improved Mixed-1

Problems	Improvement	1A	1C	1B	1C
P1	Improved	0	40	0	38
	Ties	0	0	2	2
P2	Improved	0	28	0	24
	Ties	12	12	16	16
P3	Improved	0	2	0	2
	Ties	38	38	38	38

Table 2 displays the improvement made by the improved Mixed-1 algorithm in terms of number of solutions improved and number of solutions tied. The results indicate that improved Mixed-1 significantly improves Mixed-1 and the modified Mixed-1 on problems in categories P1 and P2 (at least 40% of the roads are one-way). In fact, the improved Mixed-1 improves Mixed-1 for 68 out of 80 problems (which is 85%), and improves the modified Mixed-1 for 62 out of 80 problems (which is 78%). The results also indicate that the improved Mixed-1 improves Mixed-1 and the modified Mixed-1 very little on problems in category P3 (no more than 40% of the roads are one-way). In fact, only 2 out of 40 problems are improved. We note that for the other 38 problems in this category, the solutions obtained by Mixed-1 (hence the modified Mixed-1) are optimal. Consequently, no further improvement can be made.

3. The Mixed-2 algorithms

Frederickson [9] considered a different approach to solve the MCPP approximately. The approach is essentially the reverse procedure of the Mixed-1 algorithm. The algorithm is described in the following.

3.1. Mixed-2 algorithm (2A)

Mixed-2 algorithm consists of two phases. Phase I converts the original network into a symmetric flow network by adding some new arcs, then solving the minimal-cost flow problem on the converted flow network. Phase II transforms the symmetric network into an Eulerian one by solving the CPP over those undirected edges on the symmetric network obtained from Phase I. The performance of Mixed-2 algorithm, in the worst case, is bounded by two, (Mixed-2 solution)/ (optimal solution) $\leqslant 2$, and the bound is approachable [9].

3.2. Modified Mixed-2 (2B)

Pearn and Liu [11] observed that the artificial edges/arcs generated from the matching and minimal-cost flow solutions may form artificial cycles. Thus, a modification by removing the artificial cycles is proposed to improve the solution. Since only redundant artificial cycles are removed, (modified Mixed-2 solution) $\leqslant$ (Mixed-2 solution). Computational results show that modified Mixed-2 significantly outperform the original Mixed-2 algorithm.

3.3. Improved Mixed-2 (2C)

To further improve the solution, we propose the following modification. Let E_A be the artificial arcs added to G forming the modified Mixed-1 solution G_3. Perform the following procedure: (1) find the minimum-cost flow solution E_F over the network G_3–E_A (which is totally directed) obtained from G_3 by removing the artificial arcs E_A; (2) replace E_A by E_F to obtain the new solution $(G_3$–$E_A)\cup E_F$. For edges having two different directions in G_3, each direction will be chosen to generate a complete solution, and the best of the two is selected. Since all artificial arcs (E_A) in the modified Mixed-1 solution are removed and replaced by the minimal-cost flow solution (E_F), it is easy to see that (improved Mixed-2 solution) $\leqslant$ (modified Mixed-2 solution).

Example 3. Consider the network depicted in Fig. 3a. The depot node is 1, and arc (edge) lengths are defined as: $d(1, 3) = d(1, 5) = d(5, 2) = d(3, 4) = 1$, and $d(1, 2) = d(2, 1) = d(2, 3) = d(3, 2) = d(1, 4) = d(4, 1) = d(4, 5) = d(5, 4) = M$. Mixed-2 solution (Fig. 3b) is $6M + 4$, the modified Mixed-2 solution (Fig. 3c) is $6M + 4$, and the improved Mixed-2 solution (Fig. 3d) is $4M + 6$ which is also optimal. We note that the improved Mixed-2 solution significantly improves Mixed-2 and the modified Mixed-2 solutions. In fact, for sufficiently large M, (improved Mixed-2 Solution)/(Mixed-2 solution) = (improved Mixed-2 solution)/(modified Mixed-2 solution) $\cong 2/3$.

To compare the improved Mixed-2 with Mixed-2 and the modified Mixed-2, we run the same 120 sample problems described and tested earlier. Table 3 displays the performance comparisons among the three Mixed-2 algorithms in terms of (1) average percentage above the lower bound, (2) worst percentage above the lower bound, (3) number of problems receiving the best solutions, and (4) number of problems achieving the problem lower bounds (hence the solutions must be optimal). The results indicate that:

(a) the improved Mixed-2 improves Mixed-2 by 1.7%, 4.3%, 2.6% and improves the modified Mixed-2 by 0.9%, 1.4%, and 0.4% on problems in category 1, category 2, and category 3, respectively, in terms of average percentage above the lower bound.
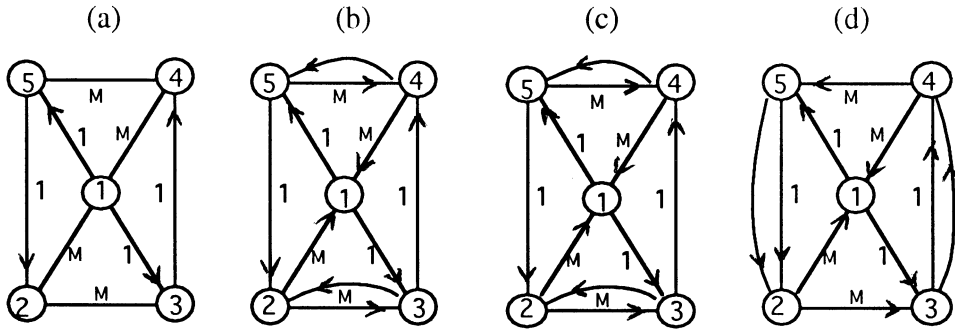
(a)   (b)   (c)   (d)

Fig. 3a–d. Example 3 and the three Mixed-2 solutions.

Table 3
Comparisons among the three Mixed-2 algorithms

Problems		2A	2B	2C
P1	Average % above the lower bound	26.79	26.05	25.12
	Worst % above the lower bound	62.67	62.67	59.41
	Number of best solutions	9	11	40
	Number of optimal solutions	0	0	0
P2	Average % above the lower bound	9.06	6.20	4.81
	Worst % above the lower bound	42.73	42.73	42.73
	Number of best solutions	3	8	40
	Number of optimal solutions	0	0	1
P3	Average % above the lower bound	3.24	1.06	0.62
	Worst % above the lower bound	18.30	3.56	3.21
	Number of best solutions	6	15	40
	Number of optimal solutions	3	7	10

(b) the improved Mixed-2 reduces the worst percentage above the lower bound by 3.3%, and 0.4% on problems in category 1, and category 3, respectively.
(c) the improved Mixed-2 improves the modified Mixed-2 for 29, 32, and 25 problems on problems in category 1, category 2, and category 3, respectively, in terms of number of best solutions.

Table 4 displays the improvement made by the improved Mixed-2 algorithm in terms of number of solutions improved and number of solutions tied. The results indicate that the improved Mixed-2 significantly improves Mixed-2 and the modified Mixed-2 on problems in all three categories. In fact, the improved Mixed-2 improves Mixed-2 for 102 out of 120 problems (which is 85%), and improves the modified Mixed-2 for 86 out of 120 problems (which is 72%). The improvement made, is significant.

4. The mixed strategies

In attempting to obtain better solutions, Frederickson [9] considered a mixed-strategy approach. The mixed-strategy approach first calls Mixed-1 and Mixed-2 algorithms to generate two

Table 4
Improvement made by improved Mixed-2.

Problems	Improvement	2A	2C	2B	2C
P1	Improved	0	31	0	29
	Ties	9	9	11	11
P2	Improved	0	37	0	32
	Ties	3	3	8	8
P3	Improved	0	34	0	25
	Ties	6	6	15	15

Table 5
Comparisons among the three mixed strategies

	P1		P2		P3	
Mixed-1	4	0	32	6	38	38
Ties	0		0		3	
Mixed-2	36	0	8	0	5	3
modified Mixed-1	5	0	26	6	38	38
Ties	2		2		7	
modified Mixed-2	37	0	16	0	9	7
improved Mixed-1	24	0	26	6	40	38
Ties	20		7		12	
improved Mixed-2	36	0	21	1	12	10
Mix-12	9	0	15	6	40	38
modified Mix-12	10	0	22	6	40	38
improved Mix-12	40	0	40	6	40	38

complete MCPP solutions, then selects the best of the two. We refer to this approach as Mix-12 algorithm. Frederickson [9] showed that the worst-case performance bound for this mixed-strategy approach, is 5/3. Since we have three Mixed-1 algorithms and three Mixed-2 algorithms, then we can consider the following three mixed strategies: (a) Mix-12, which calls Mixed-1 and Mixed-2, (b) modified Mix-12, which calls the modified Mixed-1 and the modified Mixed-2, and (c) improved Mix-12, which calls the improved Mixed-1 and the improved Mixed-2. It is easy to show that (improved Mix-12 solution) $\leqslant$ (modified Mix-12 solution) $\leqslant$ (Mix-12 solution). It is worthwhile to note that if we apply the three mixed strategies, Mix-12, the modified Mix-12, and the improved Mix-12 algorithms to the four examples described earlier, the solutions obtained are all optimal.

Table 5 displays the comparisons between Mixed-1 and Mixed-2, the modified Mixed-1 and the modified Mixed-2, the improved Mixed-1 and the improved Mixed-2, and the three mixed strategies in terms of number of best solutions (the left column), and number of optimal solutions (the right column). The results indicate that for problems in category 1, the three Mixed-2

algorithms outperform the three Mixed-1 algorithms. On the other hand, for problems in categories 2 and 3, the three Mixed-1 algorithms outperform the three Mixed-2 algorithms. For the three mixed strategies, the improved Mix-12 significantly outperform Mix-12 and the modified Mix-12 for problems in categories 1 and 2. All three mixed strategies perform equally well on problems in category 3. In fact, for 38 out of 40 problems in category 3 the solutions obtained are optimal. All the six algorithms are coded in FORTRAN programming language (with FORTRAN IV Compiler) and run on PC-486 DX-100. Each solution procedure requires less than 1 CPU second of run time on each of the 120 test problems.

5. Conclusions

The Chinese postman problem on mixed networks (MCPP) is an interesting generalization of the well-known CPP which has many real-world applications. The MCPP is NP-complete. Therefore, it is difficult to solve the problem exactly. Many heuristic solution procedures including Mixed-1, Mixed-2, modified Mixed-1, and modified Mixed-2 have been proposed to solve the problem approximately. In this paper, we considered two improvement procedures called improved Mixed-1 and improved Mixed-2 to improve the existing solutions. The two improvement procedures were tested and compared with the four existing algorithms. The computational results showed that the proposed procedures significantly improved the existing solutions.

References

[1] Edmonds J, Johnson E. Matching, Euler tours and the Chinese postman. Mathematical Programming 1973;5:88–124.
[2] Papadimitriou CH. On the complexity of edge traversing. Journal of the Association for Computing Machinery 1976;23:544–54.
[3] Kappauf CH, Koehler GJ. The mixed Chinese postman problem. Discrete Applied Mathematics 1979;1:89–103.
[4] Ralphs TK. On the mixed Chinese postman problem. Operations Research Letters 1993;14:123–7.
[5] Christofides N, Benavent B, Campos V, Corberan A, Mota E. An optimal method for the mixed postman problem. System modeling and optimization. Lecture Notes in Control and Information Sciences 1983;59:641–9.
[6] Minieka E. The Chinese postman problem for mixed networks. Management Science 1979;25:643–8.
[7] Nobert Y, Picard JC. An optimal algorithm for the mixed Chinese postman problem. Networks 1996;27:95–108.
[8] Anily S, Gendreau M, Laporte G. Two solvable cases of the mixed rural postman problem. Publication 96–35, Centre de Recherche sur les Transport, 1996.
[9] Frederickson GN. Approximation algorithms for some postman problems. Journal of the Association for Computing Machinery 1979;26(3):538–54.
[10] Brucker P. The Chinese postman problem for mixed graphs. Lecture Notes in Computer Science 1981;(100):354–66.
[11] Pearn WL, Liu CM. Algorithms for the Chinese postman problem on mixed networks. Computers & Operations Research 1995;22(5):479–89.

W.L. Pearn received his Ph.D. degree from the University of Maryland at College Park, MD, U.S.A. Currently, Dr. Pearn is a professor at the Department of Industrial Engineering and Management, National Chiao Tung University, Taiwan, ROC.

Mr. J.B. Chou received his M.S. degree from National Chiao Tung University. Currently, he is a manufacturing engineer at the Manufacturing & Material Division, Actima Technology Corporation, Hsinchu, Taiwan, ROC.