International Journal of Engineering Education Vol. 39, No. 2, pp. 397–408, 2023
Printed in Great Britain
0949-149X/91 $3.00+0.00

Integrating Social Network Analysis with Cooperative Learning in Programming Courses: A Case Study*

WEN-CHIH CHANG
International Master Program in Information Technology and Applications, National Pingtung University, No. 51, Minsheng E. Rd., Pingtung City, Pingtung County 900392, Taiwan (R.O.C.). E-mail: yilan.earnest@mail.nptu.edu.tw

Cooperative learning is an effective method of learning and is crucial for learning programming. In cooperative learning, group members cooperate with each other and teach each other. For classroom clustering methods, students are allowed to choose their own groups, or they are grouped by teachers based on experience. This study proposed a social networking analysis clustering method with an experimental group and a control group taking a freshmen programming course. A significant improvement in learning effectiveness was observed. Female students were more likely to select other female students for cooperative learning. Female students in the experimental group showed better social performance than female students in the control group. Judging from the cooperation and interaction of the overall students in the first semester and the second semester, roommates were a preferred choice as teammates.

Keywords: social network analysis; cooperative learning; programming course

1. Introduction

With advancements in technology, artificial intelligence, big data, and the Internet of Things, more software engineers are required than ever before. Program design is a key component in the training of software engineers. Beginners experience frustration when they first attempt programming. When students make mistakes, errors occur in their programs, and the programs may not work. This frustration may cause students to give up.

In addition, programming requires the use of logic and abstract memory concepts. Traditional teaching methods involve using printed books or a paper and pen. Students have difficulty focusing on class content. Therefore, repetition-based learning and peer-to-peer cooperative learning groups are required. Cooperative learning can complement class content and enhance student achievements.

2. Related Works

For programming courses, we propose a combination of social network analysis (SNA) grouping for cooperative learning and the recording of digital images to present abstract concepts, allowing students to revise and self-study. The following is a literature review on cooperative learning, SNA, and programming education.

Cooperative learning is an educational approach that emphasizes organized classroom learning activities [1]. Cooperative learning provides numerous advantages. In 1994, Johnson et al. proposed five essential elements for cooperative learning in the classroom [2, 3]: (1) positive interdependence, (2) individual and group accountability, (3) face-to-face promotive interaction, (4) teaching the students the requisite interpersonal and small group skills, and (5) group processing.

Cooperative learning presents an opportunity for university students to develop interpersonal, social, and teamwork abilities; these abilities can benefit their careers and social lives [4]. Cooperative learning and collaborative learning are the primary teaching methodologies used in educational, social, and professional contexts [5]. Generally, students are unaccustomed to studying and working cooperatively, and therefore, student cooperation and interaction may not progress as the educator intends [6].

University lecturers must foster the optimal conditions to facilitate cooperative learning activities and effective learning teams. Factors that inform the planning of cooperative learning include the time provided, student groupings, student personal characteristics, basic social skills, and the academic level of students. This research focuses on student grouping.

SNA is the process of analyzing and investigating social structures through the use of networks and graph theory [7]. Betweenness centrality can be applied to social networks [8, 9]; a higher betweenness centrality of a node indicates that more information can be passed through that node. Higher betweenness centrality nodes often play the role of a "bridge" connecting two or more small groups in an SNA graph. Milgram, a professor of psychology at Harvard University, proposed the "six degrees" theory in 1967 on the basis of an experiment where he used chains of forwarded letters to demon-

* Accepted 21 December 2022.

Table 1. Experimental and control group distribution

Group	Male	Female	Sum
Experimental Group	34	10	44
Control Group	38	6	44

strate that two unacquainted Americans could contact each other through an average of six intermediaries. Most of the transmission involved a small number of celebrities [10]. Strogatz and Watts adapted the network model in 1998 to explain the network phenomena of small societies. They described two network characteristics, namely separation coefficient and clustering coefficient [11]. The term social network was first used in an academic context by Barnes; it refers to the social relationships between people [12].

Programming curricula consist of content on syntax, programming concepts, program debugging, and problem-solving. Two educational games for teaching programming concepts in higher education programming courses have been proposed [13]. Kandin and Şendurur [14] used block-based coding instructions with a goal-based approach. Some research [15] has focused on the learning experience of female engineering students in Taiwan.

Female students constitute a minority of engineering students. For cooperative learning, female students generally learned together with other female students. This study used SNA questionnaires to collect information on every student, including female students.

This study applied SNA and cooperative learning in programming education and explored gender differences among first year students in Taiwan.

3. Research Method

This study had a mixed methods research design. The study included experimental and control groups (Table 1). Both groups received the same teaching material and teaching instruction over the semester. However, the clustering method used in the cooperative learning differed between the two groups. The experimental group was clustered using SNA, and the control group was clustered by the students themselves.

The control group is clustered via student's free will which means the control group members are acquainted with each other and they probably formed a group based on preference. Owing to the limitation of students' number in the experiment is 5 members. If the control group students are based on preference, the group member could be bigger than 5 people. The bigger control groups need to separate and even they must combine with other students that they are not familiar with. There is also the other issue of the control groups, the learning ability of the students might all high or all low learning ability. That is the difference from the experimental group and the control group.

Table 2. Class schedule for 18 weeks

Week	Subject	Remark
1	Variable, Data type	Pretest
2	Condition Command	Pretest
3–4	Loop	
5–6	Array	Cooperative work1
7–8	Pointer*	Cooperative work 2
9	Midterm Exam	
10–11	Function*	Cooperative work 3
12–13	Recursion*	Cooperative work 4
14–15	Project-tic-tac-toe	Cooperative work 5
16–17	Computer and human tic-tac-toe	
18	Final Exam	Posttest

Those in the experimental group were required to answer two questions. The first question was "Who will you choose to be your team members?" The second question was "Who will you ask when you encounter problems in learning programming?" Students could write the names of 1–3 classmates. The study applied SNA clustering based on the answers to the first question and slightly modified this based on the second question to form cooperative learning teams.

Data collection was undertaken over 18 weeks (Table 2), and the process included preparation, a pretest, team member clustering, a posttest, an analysis of questionnaire responses, and interviews. The pretest was composed of five programming questions (including questions on integers, doubles, body mass index calculation, string decomposition, and if command operation).

Because the original clustering result produced unevenly distributed groups (Fig. 1), the second question responses and pretest scores were used to further inform team selection. All teams were arranged to include both high- and low-score group students. Few middle-score students were identified, and some groups had no middle-score students. Fig. 2 presents the final clustering results.

The programming content became more abstract when functions, pointers, and recursion were the education topics. The students were all instructed to answer questions and to draw memory graphs in class with the teacher. If students were unable to answer questions or to draw or explain memory graphs, they were required to practice with their teammates. The whole team was required to pass each assignment. Through the cooperative learning, the abstract concepts became more concrete and clearer to the students.

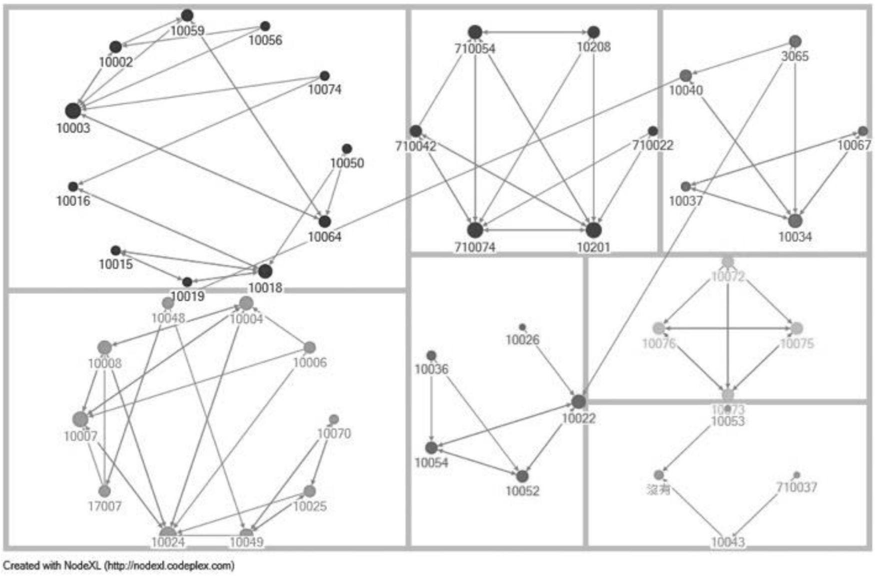


Fig. 1. Original experimental group clustering graph.

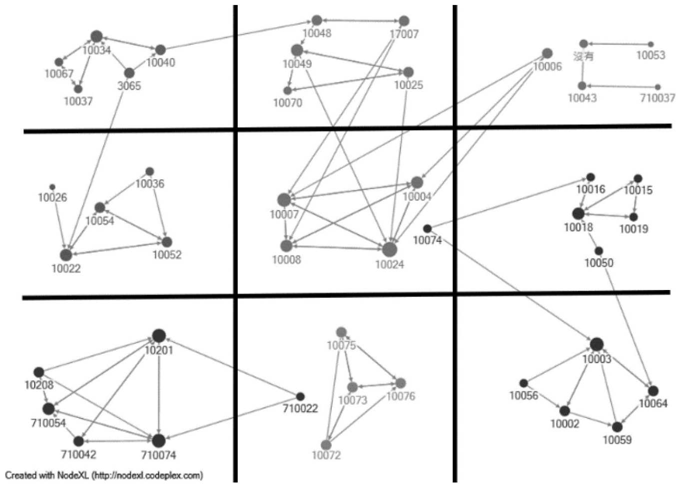


Fig. 2. Experimental group clustering graph after SNA clustering.

4. Results

4.1 Homogeneity Test

The assumptions of the statistical tests were as follows:

Null hypothesis: $H_0 : \mu_d = 0$: No significant difference was noted between the experimental and control groups.

Alternative hypothesis: $H_1 : \mu_d \neq 0$: A significant difference was noted between the experimental and control groups.

Although the pretest scores of the experimental and control groups were not significantly different ($p = 0.804$; Table 3), the posttest scores of the experimental group were significantly higher than their pretest scores ($p = 0.0001$; Table 4) and the posttest scores of the control group ($p = 0.024$; Table 5). The control group's posttest and pretest scores were not significantly different ($p = 0.465$; Table 4).

Because the statistical tests were significant, the null hypothesis was rejected; the alternative hypothesis was supported. The difference in average mean pretest scores between the experimental and control groups was 0.52; similarly, the standard deviations between the two group were 11.38 and 7.86. This implies that their learning performance levels in the pretest were similar. However, some students in the experimental group improved their average mean score from 52.93 to 63.72. This indicates that the experimental group improved their capability to learn programming skills.

4.2 SNA Subgraph for Female Students

Female students in the experimental group showed more positive and diverse social skills than female

Table 3. Pretest scores of the experimental and control groups

Group	Students	Average Score	Standard Deviation	t	P	Significance
Experimental Group	44	52.93	11.38	–0.248	0.804	No significance
Control Group	44	53.45	7.86			

Table 4. Pretest scores of the experimental and control groups

Group	Test	Students	Average Score	Standard Deviation	t	P	Significance
Experimental Group	Pretest	44	52.93	11.38	–3.796	0.0001***	No significance
	Posttest	44	63.72	16.94			
Control Group	Pretest	44	53.45	7.86	–0.737	0.465	
	Posttest	44	55.43	16.89			

Table 5. Posttest scores of the experimental and control groups

Group	Students	Average Score	Standard Deviation	t	P	Significance
Experimental Group	44	63.72	16.94	2.298	0.024*	Significance
Control Group	44	53.43	16.89			

students in the control group. In Table 6, each female student is represented by a rightmost node; each edge represents a connection with another student. In Fig. 3(a) and Fig. 3(b), female students are represented by small nodes. Generally, female students preferred to select other female students as their teammates, with the exceptions of students 910072 and 910074. These two female students were Malaysian, and they were accustomed to studying with male classmates. These students played the role of bridges; they connected other cooperative learning teammates with female Taiwanese students. In general, a bridge is a direct tie between nodes that would otherwise be in disconnected components of the graph. In both the experimental and control groups, the female Taiwanese students selected female rather than male teammates.

Some researchers [16, 17] have argued that the influence of Confucian values causes learners from a Chinese culture to have different learning and creativity styles to students influenced by Western culture [16]. To overcome the barriers to creativity and problem-solving of Chinese culture, the development of relationships between the teacher and students is crucial [18]. In this research, the experimental and control groups were all taught by the same teacher. The teacher walked around the classroom to discuss content and teach the groups individually. In addition to using traditional instruction, the teacher engaged with the students as a friend and encouraged students to learn by trial and error.

Betweenness centrality is a centrality measure of social network graphs; it is measured by the number of shortest paths between any pair of nodes that pass through the target node. For example, student 910048 connected two groups (see Fig. 3(a)) and exhibited a betweenness centrality of 97.833 (see Fig. 4(a) and Table 6).

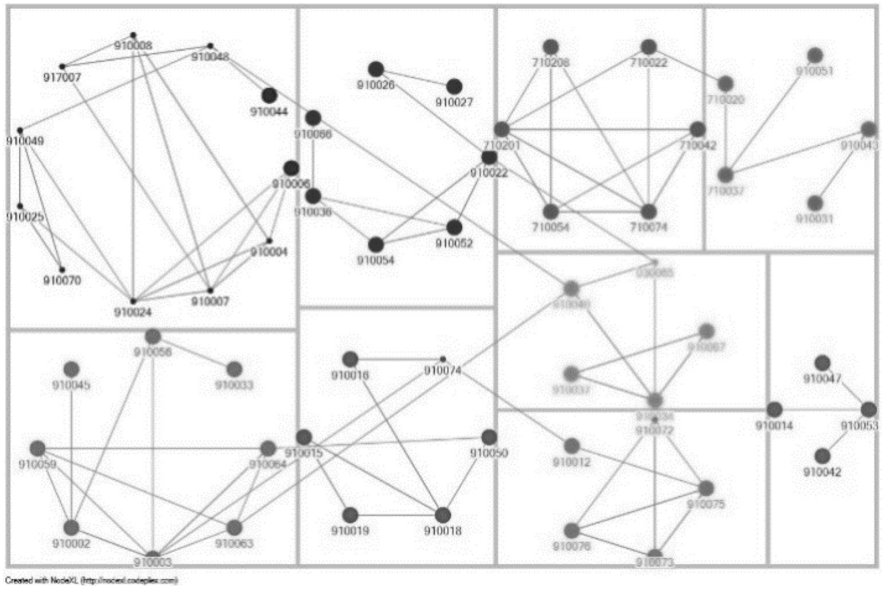
The control group exhibited higher betweenness centrality than did the experimental group. The highest betweenness centrality was 229.333 in the control group and 97.833 in the experimental group. Students in the experimental group with higher betweenness centrality than the group average of 19 generally had better academic performance. Students in the experimental group could adequately teach other programming concepts and skills (see Table 6).

In the control group, students with higher betweenness centrality than the group average of 46 generally had middle or poor academic performance. For example, the students with the highest two betweenness centralities (229.333 and 210.000) had mid-level academic performance. The ability of students in the control group to teach each other programming concept and skills was poor (see Table 7).

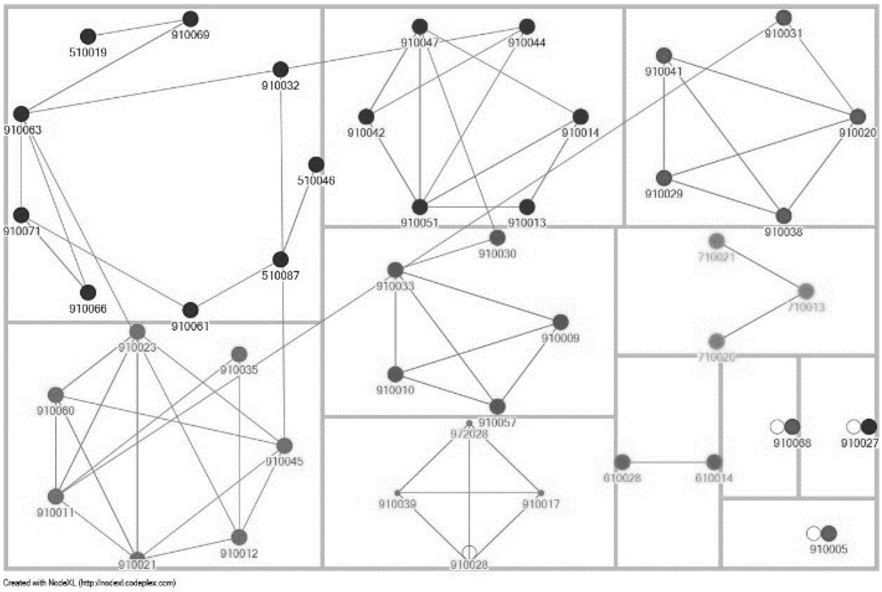
The students with high betweenness centrality in the control group were all male. According to research [19], male students tend to support female students and assist peers and juniors. However, 44.44% of students in the experimental group with high betweenness centrality were female students; 55.55% were male students. These students with high betweenness centrality and high performance could provide learning assistance in groups.

4.3 SNA for the Experimental and Control Groups

The SNA clustering method was applied to foster communication and discussion among groups. The clustering graph shown in Fig. 5(a) presents the



(a) Experimental group



(b) Control group

Fig. 3. Female student SNA graph for the experimental and control groups.

SNA graph from the first semester experiment. This study used the same SNA questionnaire for the second semester. The second semester graph is presented in Fig. 5(b). In SNA, one node represents one student, and one edge indicates that one student selected another student as a teammate. When an SNA graph has more edges, this indicates that the relevant social network has more connections. When students are not well acquainted with each other, they are too shy to select unknown or unfamiliar students as their teammates. This leads to an SNA graph with many small groups. The subgraphs are small, and the node numbers are small. In the first semester, student groups were small and had fewer edges. In the second semester, student groups were large and had more edges. After one semester of learning together, students became more familiar with one another. The second semester SNA graph has more edges, which indicates that students selected more of their classmates as teammates. In addition, no subgraphs were present that were small or isolated with only two students.

SNA is the relationship map among all the class. In Table 6 and Table 7, both the high betweenness centrality students in control group and experimen-

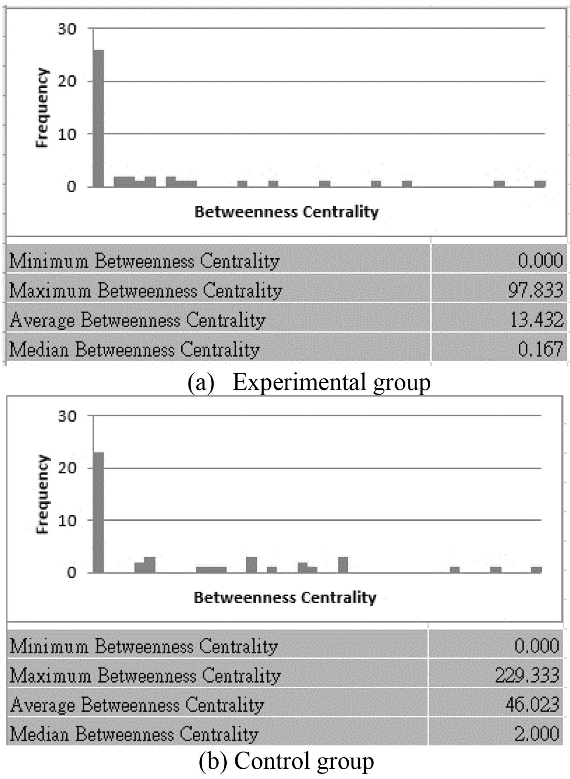


Fig. 4. Betweenness centrality of the experimental and control groups.

tal group have more male students than female students. It is a similar situation in most engineering education classes. In this research, it is successful to apply the SNA clustering with cooperative learning teaching strategy. It is suggested for engineering teachers use SNA to realize the whole teaching class. The higher betweenness centrality students with high influence than other students. If the higher betweenness centrality students have good progress from pretest to posttest. They will influence more students to study. For example, students no.910048, 910049, 910034, 917007, and 903065 in experimental group.

According to the results, dormitory roommates were the most popular choice for cooperative learning. Social network graphs show the floor plans of different dormitory buildings and off-campus area. In Fig. 5(a) and Fig. 5(b), the rectangular boxes with four decimal digits are the dormitory room numbers. For example, 1508 refers to the first dormitory for female and male students, 4425 refers to the fourth dormitory for female students, 3207 refers to the third dormitory, and 2706 refers to the second dormitory. The rectangular box with the letter "H" refers to home. Other students who were not living in the dormitory or at home were living in relatives' houses. Most students not only learned in class but also discussed and completed the assignments in their dormitories. Because the

Table 6. Students in the experimental group with higher betweenness centrality 19 than the average betweenness centrality of 13.432

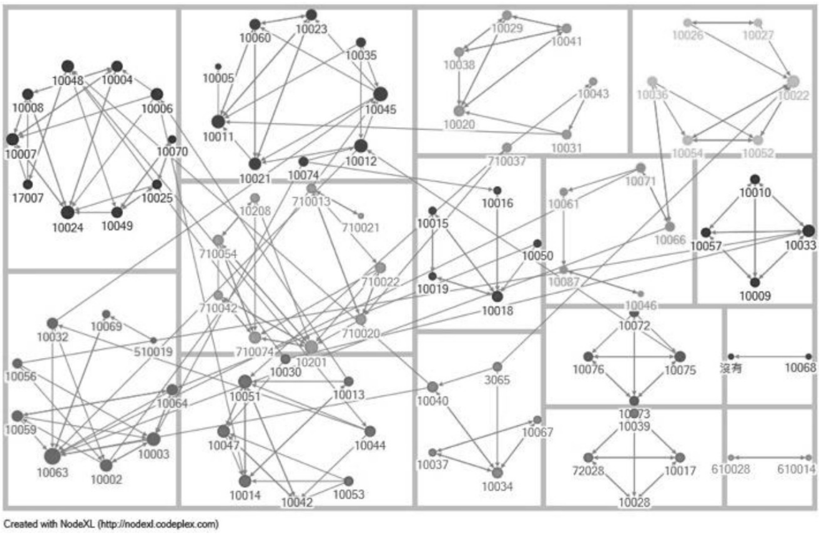
Vertex	Subgraph	Betweenness Centrality	Gender	Pretest score	Posttest score
910006		22.167	Male	60	66
910024		19.167	Female	60	80
910049		51.167	Female	60	83
917007		40.833	Female	60	81
910048		97.833	Female	43	78
910022		63.000	Male	28	6
910034		34.000	Male	36	68
910040		90.000	Male	60	60
903065		70.000	Male	36	72

four dormitory buildings were located near each other, dormitory-dwelling students could easily communicate and teach each other in person. Students living at home or in their relatives' houses were often forced to return home earlier. They could only discuss assignments using apps or websites, such as LINE, Instagram, Facebook, Microsoft Teams, and Google Meet.

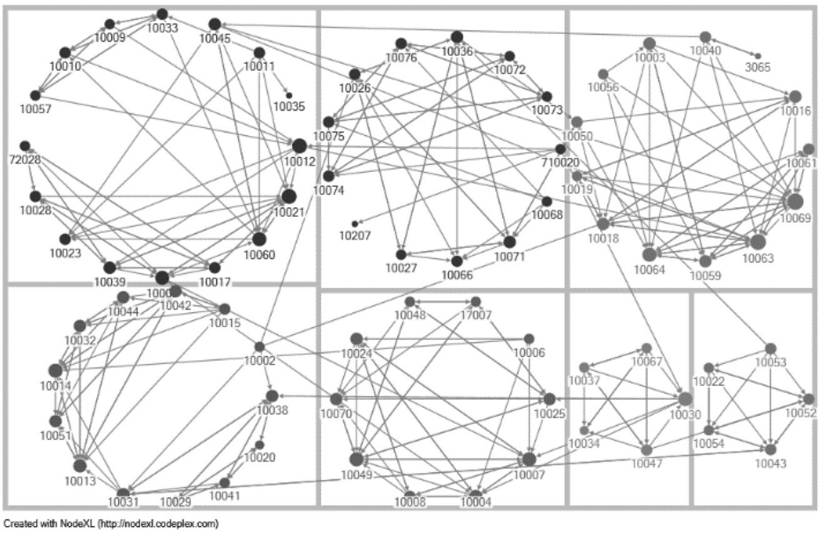
The Social Network Analysis for the first semester student groups had fewer edges. Fig. 5(b) demonstrates that the second semester student

Table 7. Students in the control group with higher than the average betweenness centrality of 46.023

Vertex	Subgraph	Betweenness Centrality	Gender	Pretest score	Posttest score
910061		58.000	Male	43	50
910071		84.000	Male	60	84
910063		190.000	Male	60	82
910020		84.000	Male	60	57
910051		114.500	Male	58	67
910042		66.000	Male	48	72
910047		130.500	Male	58	36
910044		210.000	Male	60	62
910032		229.333	Male	60	64
910045		131.667	Male	60	6
910012		92.167	Male	60	69
910021		59.333	Male	60	58
910011		132.500	Male	55	34
910031		108.000	Male	43	53
910033		84.000	Male	60	63
910030		108.000	Male	60	32



(a) First semester



Second semester

Fig. 5. Clustering distribution of the two groups.

groups had more edges. These findings indicate that students preferred to select roommates as their teammates. For example, five classmates (10020, 10029, 10031, 10038, and 10041) selected roommates as their teammates.

The use of cooperative learning facilitated more connections among classmates who had not been familiar with one another in the first semester. The width of the links of each node indicates the discussion frequency of the students. It depicts the frequency of discussion among the students over 1 week. For the second semester, each student group became more united. Almost all students were linked to the others in their group; the edges and the nodes resembled a complete graph.

The maintenance of groups with the same members over time led to the development of feelings of belonging and improvements in social skills [20]. For Johnson et al. [21], groups that stay together for at least a year with the same members and whose main objective is to give mutual support and help each other foster favorable social and cognitive development. This viewpoint could explain our findings. In particular, it may explain the progress in discussion frequency and the student clustering edge numbers. Programming learning is suitable for cooperating learning and learning by doing. It is similar to most engineering teaching subjects. The edge connections of the first semester are more complex than the second semester. It represents students in cooperative learning have foster favorable social development. Comparing the research clustering method and students' will clustering, the social network analysis clustering promotes students better understanding and academic performance in this research.

This programming course was a basic foundational course for freshmen. Therefore, it was critical to cultivate the student communication and engagement through cooperative learning.

Students could choose their teammates in the questionnaire. This approach was used to investigate the reasons for students selecting certain teammates. At the beginning of the questionnaire, the reasons why students selected teammates were collected. Each student can choose three teammates, and they could write the designed 10 reasons or write a new reason in the questionnaire sheet. It presents the preferred characteristics in teammate selection. The three most popular characteristics were "willingness to help me" with 54 votes, "enthusiastic" with 38 votes, and "easygoing" with 37 votes. Another characteristic with more than 30 votes was seriousness about teaching. Students preferred teammates who taught with a serious attitude. The most unexpected result for teammate selection was "excellent score," which only received 20 votes. The results indicate that high academic achievement was not necessarily an expectation for teammates.

4.4 Content Analysis of Programming Exam Videos

We chose the final programming exam video for recording student answers. This video was analyzed through content analysis and behavior sequence analysis.

Content analysis relates to investigations of documents and methods of communication. It may involve texts in various formats, pictures, audio, or video. Some scientists apply content analysis to investigate patterns in communication in a replicable and systematic manner. One of the advantages of using content analysis for analyzing social phenomena is its noninvasive nature. Researchers can simulate social experiences, collect survey questionnaire data, or record videos [17, 18] before investigating related patterns.

With the increase in common computing facilities such as computers and computer-assisted technologies, an increasing number of computer-based methods of analysis are used in content analysis. Answers recorded through videos, open ended questions, Wikipedia, discussions, medical records, or systematic observations in experiments can be subject to systematic analysis of textual data or video content analysis. The communication content can be transformed to machine-readable text. The input data are analyzed for frequencies and coded into categories for analysis as patterns, types, or code.

Some computer-assisted methods or technologies can speed up analysis of large digital data sets. However, human coders are still crucial for content analysis because they are often better able to identify nuanced and latent meanings in text.

To investigate the student midterm exam results for the programming course designed in this research, we observed the processes of problem-solving behavior in our recorded videos. We organized and summarized the content into five codes for analysis based on our programming exam videos. The programming coding scheme is divided into five codes (see Table 8). Each code represents a type of knowledge construction evident in the video content analysis of programming.

Coding Scheme Definition:

C1: Coding (including debugging and copying and pasting the code, which we needed to use repeatedly to compile and to test the program.

C2: Searching for references (including references on the Internet, assignments that were previously uploaded to the platform, reference materials and file on the platform, or recorded teaching videos.

C3: Viewing or reading the questions, examining the code or program (switching the program to compare, indent, or view debugging information; viewing the execution results.

C4: Thinking (thinking about how to code or what to do next)

C5: Other (e.g., asking the teacher questions on the platform, opening a folder or file, saving a file, saving as a new file, switching windows quickly

Table 8. The programming coding scheme table

Code	**Phase**	**Description**
C1	Coding/Debug	The process of students writing programs or debugging.
C2	Search for information	Search for information on the Internet, watch the recorded teaching videos, or reference previous assignment programs in this programming course.
C3	Review questions/ Review code/ Debug information	Review or check the exam questions and their own codes and debug information.
C4	Thinking	Think about how to code or what to do next.
C5	Others	Content other than the four described codes.

for no obvious reason, and tasks not involving the four defined codes.

The difference between C1 and C3 was that C1 related to writing or testing code and C3 concerned checking the accuracy of code and debugging information. The main difference between C4 and C5 was that C4 usually referred to thinking for a long time without obvious actions. C5 indicates some actions that have nothing to do with the defined behaviors or some unrelated behaviors.

4.5 Small-Scale Analysis: Behavior Analysis of Online Problem-Solving Videos based on Lag Sequence Analysis and Content Analysis

Content analysis [19] is an ideal approach for this study because it can effectively analyze the students learning program status. Understanding programming-related problem-solving processes [22] of students is important, which can help teachers realize the continuous programming coding status. A sequential analysis was applied [23–29] to visualize and analyze learners' behavioral patterns in different clusters. According to the analysis results, a deeper understanding of learning processes could be obtained, and behavioral patterns could be determined if, for example, a behavioral sequence from behavior A to behavior B in the entire learning or recording process of the clusters reached statistical significance.

In this study, lag sequential analysis was used to analyze students' answers to programming problem in the final exam. An analysis of problem-solving-related video sequences can provide a deeper understanding of how students solve problems.

Studies [19, 32] have investigated the relationship between sex and achievement in computer programming. The results [30] indicated significant advantages in prior conceptual and strategic knowledge among male students. Male students exhibited higher scores in conceptual knowledge and strategic knowledge than did female students in the programming course. Female students were more successful than male students in their initial programing status and in their syntactic programming knowledge development. One study [19] reported that male students performed better than female students in understanding concepts, working in programming environments, and debugging. That result was based on experimental results and a questionnaire. However, a video analysis and codes for the educational theories were not used. Therefore, this study applied content analysis and lag sequential analysis to determine sex-related differences.

In this small-scale analysis, 18 students were evenly divided into a high-score group, a middle-score group, and a low-score group. All content

Table 9. The programming code distribution of all students

Code	Count	Percentage %
A sample of 18 students		
C1	3128	37%
C2	2768	32%
C3	1319	15%
C4	1004	12%
C5	395	4%
A sample of 9 male students		
C1	1605	36%
C2	1488	34%
C3	576	23%
C4	546	12%
C5	208	5%
A sample of 9 female students		
C1	1580	37%
C2	1280	30%
C3	773	18%
C4	428	10%
C5	187	5%

analysis results are listed for the male students. We video-recorded the code described in Table 9. Female students spent less time answering the programming questions that are presented in Table 9. Male students spent more time on C3 (review question, coding and debugging information) than female students did.

The results in Table 9 indicate that compared with male students, female students had better results for C1 in their programming coding video. The finding suggests that female students were more capable of coding and debugging than their male peers. One study reported that male students outperformed female students in understanding concepts, working in programming development environments, and fixing bugs [19]. This viewpoint helped to elucidate our findings. It is suggested that to arrange some good academic performance male students help female students in coding and debugging programming practice.

The sequential analysis was further explored using the five behavior codes (C1, C2, C3, C4, and C5) of all the students and both the female and male clusters. The adjusted residuals table for each cluster of programming-related problem-solving data is presented in Table HH. Each row pertains to initial behaviors, and the columns refer to subsequent behaviors. *Z* scores [31] greater than 1.96 represent the continuity from one state to another in behavioral sequences that achieved statistical significance ($p < 0.05$) [25]. In accordance with Table 3, the behavioral patterns are presented in Fig. 6, 7, and 8. Behaviors are presents as rectangles, and arrows point to other behaviors. The significant sequence is attached with a z score,

Table 10. Adjusted residual table for three clusters of programming-related problem-solving behaviors

Z	C1	C2	C3	C4	C5
All students					
C1	–52.633	–1.718	10.665*	–3.728	–6.189
C2	7.480*	–12.190	–1.103	–7.558	–2.207
C3	6.085*	–6.162	–58.220	–1.533	–0.965
C4	0.610	–5.407	3.788*	–13.970	–2.493
C5	–4.398	–1.153	7.772*	–3.598	–4.266
Male					
C1	–38.134	–1.633	8.863*	–2.998	–4.585
C2	5.318*	–10.355	–1.560	–4.620	–1.880
C3	3.990*	–4.594	–43.796	–1.668	–0.182
C4	–0.417	–3.272	3.836*	–8.114	–2.654
C5	–3.171	–0.842	5.625*	–2.761	–3.193
Female					
C1	–36.275	–0.679	6.160*	–2.405	–4.150
C2	5.129*	–6.923	–0.017	–5.797	–1.279
C3	4.574*	–4.349	–38.499	–0.310	–1.281
C4	1.051	–4.065	1.466	–11.973	–0.957
C5	–3.056	–0.834	5.361*	–2.310	–2.833

* $p < 0.05$.

which are corresponded to the z scores in Table 10. The bigger the number, the bigger the impact.

A sequence analysis for a sample of 18 students was conducted, as presented in in Fig. 10; this figure details the behavior sequences of students solving programming problems. A loop behavior was noted between C1 and C3.

According to the event transition diagram in Fig. 10, the link from C1 to C3 was the strongest (10.665). This indicates that both female and male students prefer coding to transferring and executing programs. The link from C2 to C1 was 7.48. This indicates that most students determined the requirements to solve a problem and gathered the necessary information. No obvious link was observed from C1 to C2. This indicates that students did not need to search for additional information when coding.

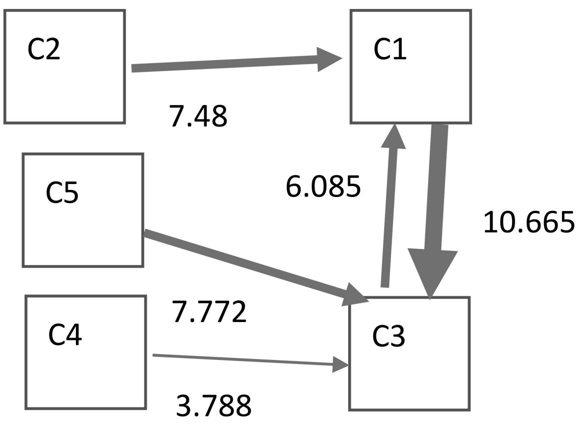


Fig. 6. Event transition diagram of a sample of 18 students.

We conducted gender-related investigations through a sequence analysis of a sample of nine female students (Fig. 7) and nine male students (Fig. 8).

The sequential patterns in Fig. 7 and Fig. 8 indicate that compared with female students, male students had more sequential links between "review questions, coding, and debugging information" (C3) and "thinking" (C4). Female students only had the following links: C2–>C1, C1–>C3, C3–>C1, C5–>C3. Comparing female students' analysis in Fig. 7, male students have smaller z score from C3 to C1.

Concerning the bidirectional connection of "Coding and debugging" and "Review questions, coding, and debugging information," male and

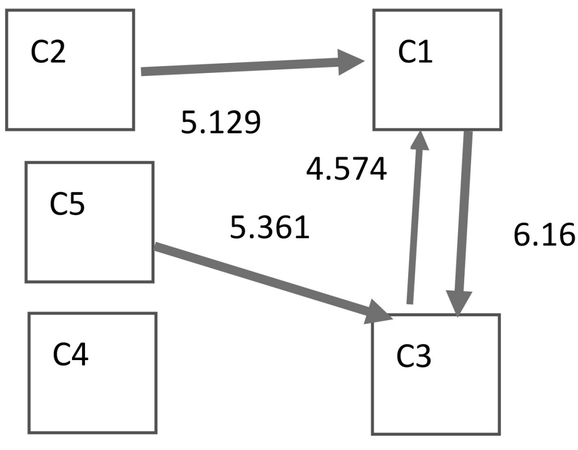


Fig. 7. Event transition diagram for a sample of 9 female students.

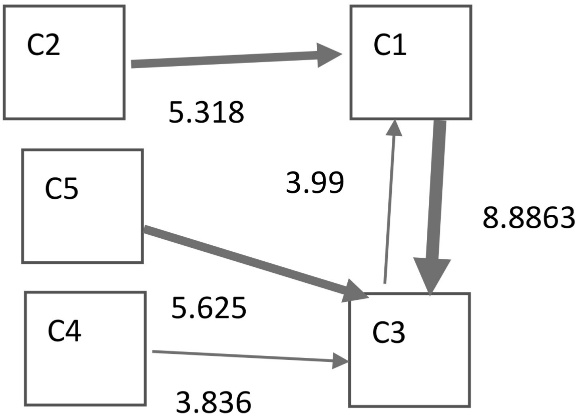


Fig. 8. Event transition for a sample of nine male students.

female students both exhibited significant bidirectional connections. A sequential analysis revealed that male students exhibited stronger connections (C1–>C3) than did female students. This indicates that male students had relatively higher a score behavior in C2–>C1(5.318>5.129), C1–>C3(8.8863>6.16) and C5–>C3(5.625>5.361) than female students. This indicates that male students tended to code and analyze the programming instructions and then use trial-and-error strategies to test the programs, which generate the output and rewrite the programs according to the debug and coding.

5. Conclusion

In this research, an SNA-based clustering approach was applied to cooperative learning in a programming course. The experimental group was formed using an SNA-based clustering approach, and the control was formed by free will. The pretest score of the experimental and control groups were not significantly different. The posttest scores of the experimental and control groups were significantly different. In this research, female students in the experimental group were more active and better at social interactions than were those in the control group. From the first semester to the second semester, students became more active and communicated more with the other students. In addition, dormitory roommates became primary cooperators. Content analysis indicates that female students had a higher C1 value in their programming coding video. The finding suggests that female students were more likely to have coding debugging capabilities than their male peers. The sequential analysis revealed that compared with female students, male students had better directional connections in "Review questions, coding, and debugging information."

Acknowledgements – The author would like to thank MOE (Ministry of Education) Teaching Practice Research Program Project PEE1101188 financial support. This manuscript was edited by Wallace Academic Editing.

References

1. R. M. Gillies, Cooperative Learning: Review of Research and Practice, *Australian Journal of Teacher Education*, **41**(3), pp. 39–51, 2016.
2. D. W. Johnson, R. T. Johnson and E. J. Holubec, The nuts and bolts of cooperative learning, *Interaction Book Co.*, 1994.
3. D. Johnson and R. Johnson, Learning together and alone, cooperative, competitive, and individualistic learning. Needham Heights, MA: *Prentice-Hall*, 1999.
4. S. Mendo-Lázaro, B. León-Del-Barco, E. Felipe-Castaño, M. I. Polo-Del-Río and D. Iglesias-Gallego, Cooperative Team Learning and the Development of Social Skills in Higher Education: The Variables Involved. *Frontiers in Psychology*, **9**, 1536, 2018.
5. I. St-Pierre and M. Richer, Cooperative education in schools: the quebec case, *Educ. Invest. Arbitrada*, **12**, pp. 109–116, 2008.
6. C. Buchs and F. Butera, Cooperative learning and social skills development, in *Collaborative Learning: Developments in Research and Practice*, New York, NY: Nova Science, pp. 201–217, 2015.
7. O. Evelien and R. Ronald, Social network analysis: a powerful strategy, also for the information sciences, *Journal of Information Science*, **28**(6), pp. 441–453, 2002.
8. L. Freeman, A set of measures of centrality based on betweenness, *Sociometry*, **40**(1), pp. 35–41, 1997.
9. W. C. Chang, H. W. Lin and L. C. Wu, Applied Social Network Analysis to Project Curriculum, *Proceeding of the 6th International Conference on Networked Computing and Advanced Information Management*, pp. 710–715, 2010.
10. S. Milgram, The Small World Problem, *Psychology Today*, **2**, pp. 60–67, 1967.
11. D. Watts and S. Strogatz, Collective dynamics of 'small-world' networks, *Nature*, **393**, pp. 440–442, 1998.
12. J. A. Barnes, Class and Committees in a Norwegian Island Parish, *Human Relations*, **7**, pp. 39–58, 1954.
13. D. Zhao, C. H. Muntean, A. E. Chis, G. Rozinaj and G.-M. Muntean, Game-Based Learning: Enhancing Student Experience, Knowledge Gain, and Usability in Higher Education Programming Courses, in *IEEE Transactions on Education*, pp. 1–12, 2022.
14. E. Kandin and E. Şendurur, The effects of goal-based scenarios used for programming education of 5th graders, *Interactive Learning Environments*, pp. 1–21, 2022.
15. P-.N. Chou and W-.F. Chen, Female engineering students' perceptions of college learning experiences: A qualitative case study in Taiwan, *International Journal of Engineering Education*, **31**(1), pp. 2–11, 2015.
16. K. H. Kim, Learning from each other: creativity in East Asian and American Education, *Creativity Research Journal*, **17**(4), pp. 337–347, 2005.
17. J. A. Goncalo and B. M. Staw, Individualism–collectivism and group creativity, *Organizational Behavior and Human Decision Process*, **100**, pp. 96–109, 2006.
18. C. Zhou and J. Shi, A Cross-Cultural Perspective to Creativity in Engineering Education in Problem-Based Learning between Denmark and China, *International Journal of Engineering Education*, **31**, pp. 12–22, 2015.

19. N. D. Gunasekara, Gender Differences in Learning Programming: An Analysis by Students' Perception, *International Journal of Research and Innovation in Social Science*, **5**(4), pp. 149–155, 2021.
20. F. R. Javier and M. G. Antonio, *El aprendizaje cooperativo: modelo pedagógico para educación ísica,* Retos 29, pp. 201–206, 2016.
21. D. W. Johnson, R. T. Johnson and E. J. Holubec, *El Aprendizaje Cooperativo en el Aula*, Buenos Aires: Paidós, pp. 1–39, 1999.
22. H. T. Hou, Y. T. Sung and K. E. Chang, Exploring the behavioral patterns of an online knowledge sharing discussion activity among teachers with problem-solving strategy, *Teaching and Teacher Education*, **25**(1), pp. 101–108, 2009.
23. R. Bakeman and J. M. Gottman, *Observing interaction: An introduction to sequential analysis* (2nd ed.). UK: Cambridge University Press, 1997.
24. H. T. Hou, Applying lag sequential calculation and social network analysis to detect learners' behavioral patterns and generate automatic learning feedback-scenarios for educational MMORPG games. *In Proceeding of the IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning*, pp. 213–215, 2010.
25. H. T. Hou, Integrating cluster and sequential analysis to explore learners' flow and behavioral patterns in a simulation game with situated-learning context for science courses: A video-based process exploration, *Computers in Human Behavior*, **48**, pp. 424–435, 2015.
26. H. T. Hou and S.-Y. Wu, Analyzing the social knowledge construction behavioral patterns of an online synchronous collaborative discussion instructional activity using an instant messaging tool: A case study, *Computers & Education*, **57**(2), pp. 1459–1468, 2011.
27. P. C. Lin, H. T. Hou, S. M. Wang and K. E. Chang, Analyzing knowledge dimensions and cognitive process of a project-based online discussion instructional activity using Facebook in an adult and continuing education course, *Computers & Education*, **60**(1), pp. 110–121, 2013.
28. H. T. Hou, A case study of online instructional collaborative discussion activities for problem-solving using situated scenarios: An examination of content and behavior cluster analysis, *Computers & Education*, **56**(3), pp. 712–719, 2011.
29. H. T. Hou, K. E. Chang and Y. T. Sung, Analysis of Problem-Solving Based Online Asynchronous Discussion Pattern, *Educational Technology & Society*, **11**(1), pp. 17–28, 2008.
30. P. Y. Mohammed and M. B. Raewf, The Content Analysis of the Research Papers on Language Learning Strategies, *Qalaai Zanisty Journal*, **3**(4), pp. 904–911, 2021.
31. P. D. Allison and J. K. Liker, Analyzing sequential categorical data on dyadic interaction: A comment on Gottman, *Psychological Bulletin*, **91**(2), pp. 393–403, 1982.

Wen-Chih Chang is an associate professor of International Master Program in Information Technology and Applications, National Pingtung University, Pingtung, Taiwan. He received his BS, MS and PhD in the computer science and information engineering department in Tamkang University, Taiwan. Dr. Chang's research interests include social network analysis, e-learning and computer assisted learning.