

## 파라메트릭 매크로기법에 의한 부재생성 및 부재배치방법에 관한 연구

한 창 봉\* · 박 제 응\*\*

### A Study on the Generating Piece and Arrangement Method Using the Parametric Macro

C. B. Han\* · J. W. Park\*\*

Key Word : INF 파일(부재의 배치를 위한 선택된 부재들의 정보파일), Autorun(자동 스트리핑 프로그램), STRIP(부재를 강재에 배치하는 작업), ENDCUT(부재 모서리 부분), TRIBON(조선전용 CAD S/W), \*.FIL 파일(ENDCUT에 의하여 정의된 부재 모서리부분의 형상)

#### ABSTRACT

large size shipyards have performed the generating of piece and the nesting using shipbuilding-oriented CAD program, which requires higher skilled operators and lots of costs. So, medium and small size shipyards have pursued different types of piece generation and nesting (arrangement method). This research is to develop a program for the generating of pieces with parametric macro method and a STRIP program for the nesting of pieces by PC, which can be applied in such shipyards.

#### 초 록

대형 조선소에서는 선박에 사용하고 있는 수많은 부재들은 고가이고, 고급기술을 필요로 하는 WORK-STATION급에서 구동되는 선박전용프로그램 등에 의하여 부재생성 및 네스팅을 하고 있다. 그러나 중소형 조선소의 경제적, 기술적인 사정으로 고가의 장비와 고급인력의 확보가 어려운 실정이다. 따라서 본 연구에서는 널리 보급되어 있는 PC에서 사용 가능한 파라메트릭 매크로에 의한 부재생성과 부재의 네스팅 프로그램을 개발하였다.

#### 1. 서 론

선박설계에 있어 생산설계 단계에서 부재생성 및

네스팅 단계에서 여러 프로그램들을 사용하고 있으며, 대형 조선소에서는 현재 많이 사용하는 TRIBON과 같은 조선 전용 프로그램을 활용하여

\* 조선대학교 대학원(Chosun University, Graduate School)

\*\* 조선대학교 선박해양공학과(Chosun University, Dept. of Naval Architecture and Ocean Eng.)

네스팅 작업을 하고 있다. 그러나 이러한 네스팅 프로그램들은 워크스테이션(work station)이나 마이크로 컴퓨터에서 사용되는 것이므로 사용법을 배우는데도 많은 시간이 걸리고, 장비의 고가로 중·소형 조선소에서 이러한 프로그램들의 사용은 현재까지 어려운 실정이다. 따라서 중·소형 조선소의 조선 기술 향상 및 선박 품질을 높이기 위한 목적으로 PC에서 사용가능<sup>(6)</sup>한 내부재 부재 생성과 생성된 부재의 네스팅 프로그램을 개발하기에 이르렀다.

2. 프로그램 구성도

Fig. 1은 본 연구에서 개발된 프로그램의 구성도로써 ENDCUT-DEFINE에의해 구축된 매크로를 이용하여 부재를 생성한 후, 네스팅하게 된다.

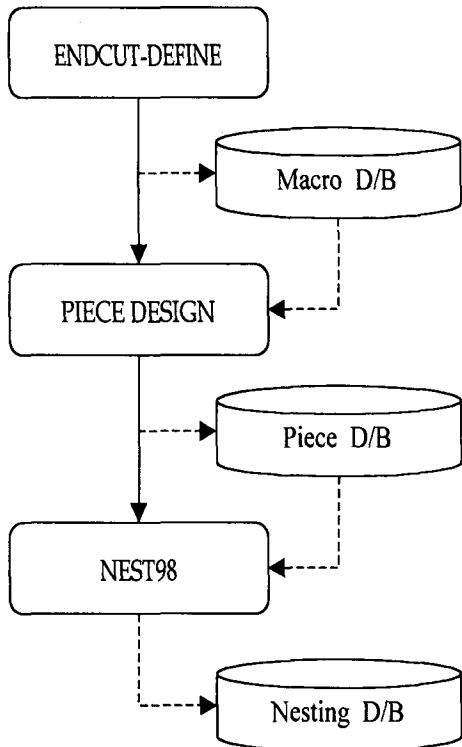


Fig. 1. Flow chart of program

3. 부재외형 매크로정의

부재를 형성하기 위해서는 부재외형을 매크로로 정의하고, 정의된 부재형상의 매크로는 각각의 ID로 저장된다. 부재를 생성하려할 때 해당되는 ID를 찾아 부재를 생성하고 네스팅하기 위하여 \*.PTC파일을 생성하게 하였다. 부재형상 정의는 사용자가 직접 매크로를 만들어 사용하도록 준비한 기능으로, 각각의 조선소에 따라 같은 형태의 형상이라도 그 수치 값이 조금씩 차이가 있을 수 있고, 또한 이 차이를 모두 라이브러리화 한다는 것은 프로그램이 너무 방대해지기 때문에 사용자가 구성하여 사용할 수 있도록 하였다. 그 흐름도는 Fig. 2와 같다.

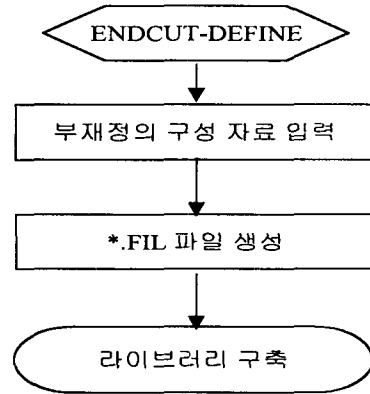


Fig. 2. Flow chart of ENDCUT-DEFINE

부재외형을 정의하기 위하여 입력되는 파라미터의 예는 Table 1과 같다

Table. 1. Example for Parametric characteristic of ENDCUT-DEFINE

ID-NAME	E C 1504	PROFILE	F B	
SIZE	150 × 0 × 0 × 0			
NO	DR	RADIUS	DX(C X)	DY(C Y)
1	3	0.0	120.0	30.0
2	9	35.0	-35.0	0.0
3	5	0.0	-85.0	35.0
4	3	0.0	0.0	50.0
5	1	0.0	30.0	0.0
6	9	35.0	0.0	35.0

Table. 2. Contents of ENDCUT-DEFINE

ID-NAME : 저장될 FILE NAME
PROFILE : 부재의 PROFILE(A N, F B, B K, …… )
SIZE : WEB-길이×FLANGE-폭×WEB-두께×FLANGE-두께
NO : 엔티티가 그려지는 순서
DR : 엔티티의 진행 방향, LINE : 1~8, ARC인 경우 : 9
D X, D Y : 상대좌표
C X, C Y : 호의 중심 좌표

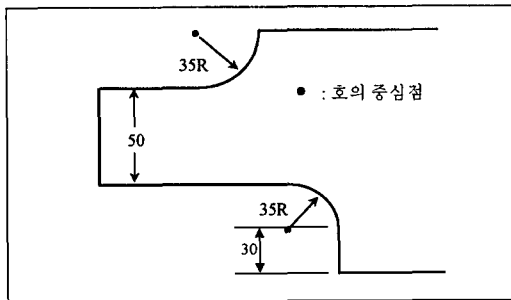


Fig. 3. Form of ENDCUT

Table. 1과 같이 부재외형의 좌표값들은 기준점에서 상대좌표 값으로 입력되며 정의된 정보는 STANDARD 디렉토리에 "FILE NAME : EC1504"으로 저장되고, 입력받은 데이터의 내용은 Table. 2와 같다.

그리고 입력받은 데이터 값은 데이터베이스에 저장되고, 위의 정보로 입력받은 부재외형의 모서리 부분의 형상은 Fig. 3과 같다.

Fig. 3과 같이 Flat-Bar의 형상은 좌측에 붙는 것을 정의하고, 우측에 붙을 경우는 좌표변환하여 부재를 설계하게 된다.

#### 4. 부재 생성 프로그램

부재생성 프로그램은 PC전용 CADRA 프로그램에서 생성된 부재를 확인할 수 있도록 하였다. 그리고 CADRA 프로그램에서 부재를 생성할 때, 간편하고 쉽게 생성하기 위하여 대화형 시스템과 배치형식 시스템을 병용하여 개발하였다.

PC-CADRA 프로그램에서 부재를 생성할 때

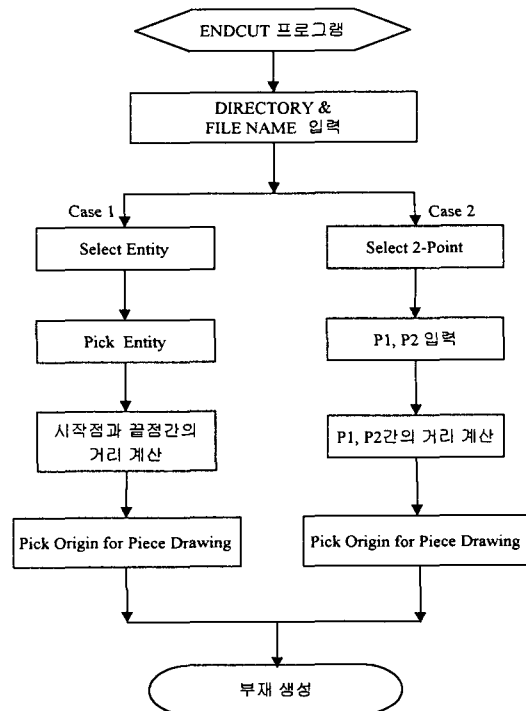


Fig. 4. Flow chart of ENDCUT program

"ENDCUT-DEFINE"에서 정의된 라이브러리를 이용하여 부재를 생성하게 된다. 이 기능은 생성할 부재의 폭을 "ENDCUT- DEFINE"에서 받아들이고 부재의 파라미터를 입력받아 그리게 하였다.

이 기능의 흐름도는 Fig. 4와 같다.

생성할 부재의 형상을 "ENDCUT-DEFINE"에서 정의한 부재 외곽선의 형태를 먼저 입력받고, 설계할 부재가 저장될 디렉토리와 부재 이름을 입력한 뒤, 부재의 파라미터를 입력받아 부재를 생성하게 된다.

필요한 정보를 입력받고, 그려질 부재의 길이를 입력하기 위한 방법은 엔티티를 선택하여 길이를 입력받는 방법과, 두 점으로 부재의 길이를 입력받는 두 가지 방법으로 나누었다.

##### 4.1 case 1 : 엔티티를 선택하는 경우

부재를 생성하기 위하여 엔티티를 선택하면, 엔티티의 길이를 계산하여 엔티티의 길이와 같은 길이로 부재를 생성하게 된다. 생성된 부재의 형상을 보면

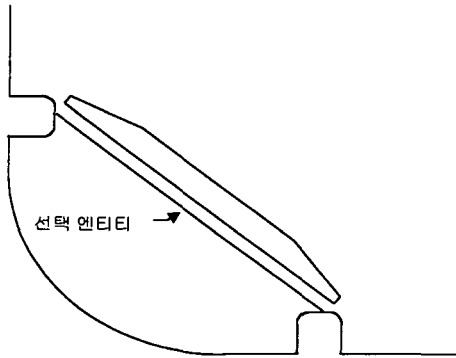


Fig. 5. Case of selected entity for generating piece

Fig. 5와 같다.

4.2 case 2 : 두 점을 선택하는 경우

두 점을 선택하면 두 점간의 거리를 계산하여 부재의 길이를 두 점이 이루는 거리로 하여 부재를 생성하게 된다. 생성된 부재의 형상을 보면 Fig. 6과 같다.

이와 같이 생성된 부재들은 데이터베이스 등록하고 스트리핑에 사용할 수 있도록 하나의 파일인 \*.PTC파일을 생성한다.

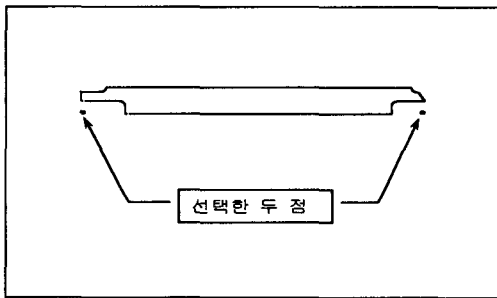


Fig. 6. Case of selected two point for generating piece

5 생성된 부재확인

파라메트릭 매크로에 의하여 생성된 부재를 확인하기 위한 기능인 "RDPTC"는 선택된 부재의 각

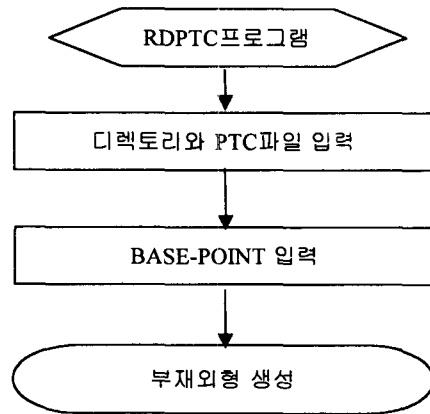


Fig. 7. Flow-chart of RDPTC program

엔티티 좌표값들을 임의의 기준점으로부터 상대 좌표값을 계산하여 그린다.

프로그램의 흐름도는 Fig. 7과 같다.

생성된 부재를 확인하기 위하여 디렉토리와 부재이름을 먼저 입력한다. 디렉토리가 "C:\CHOSUN\PIECE" 이고 파일 이름이 "TEST.PTC"인 경우는 다음과 같다.

```
C : \ CHOSUN \ PIECE \ TEST01
```

부재는 LINE과 ARC로 이루어져 있으므로 PTC 파일은 LINE과 ARC를 구분하여 기록을 하였다. \*.PTC 파일의 정보는 Table. 3과 같다.

Table. 3. Example of test01.ptc file

LINE, 30001, 50.000, 0.000, 450.000, 0.000,
ARC, 30002, 50.000, 500.000, 0.000, 1.570, 3.141,
LINE, 30003, 50.000, 50.000, 550.000, 100.000,
LINE, 30004, 550.000, 100.000, 550.000, 200.000,
LINE, 30005, 550.000, 200.000, 0.000, 200.000,
LINE, 30007, 0.000, 200.000, 0.000, 50.000,
ARC, 30006, 50.000, 0.000, 0.000, 0.00, 1.570,
END,

Table. 3과 같은 PTC파일을 CADRA에서 읽어 들일 때 각각의 좌표는 상대좌표로 그려질 기준점을 입력하면 상대좌표로 다음과 같이 입력받게 하였다.

LINE인 경우

LINE	30001	50.000	0.000	450.00	0.000
엔티티 구분	엔티티 순서	Xstart	Ystart	Xend	Yend

ARC인 경우

ARC	30002	50.000	500.000	0.000	1.570	3.141
엔티티 구분	엔티티 순서	ARC 반지름	Xcenter_p	Ycenter_p	θstart	θend

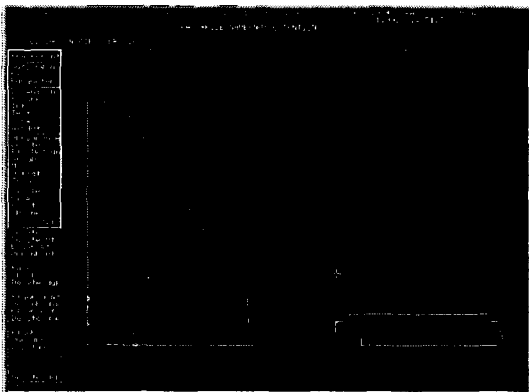


Fig. 8. Identification of pieces

입력받은 정보로 LINE과 ARC를 그리게된다. 부재의 정보를 읽어들이어 그려진 그림은 Fig. 8과 같다.

### 6. 부재 관리

데이터베이스는 크게 두 부분으로 나누어 구축되었다. 먼저, 책의 목차부분과 비슷한 인덱스 파일인 \*.IDX파일과, 책의 내용에 비교할 수 있는 데이터 파일인 \*.DAT파일로 구분하였다. 데이터베이스의 이름이 NAVAL이라면 인덱스 파일과 데이터 파일의 이름은 Table. 4와 같다.

Table. 4. Example of NAVAL database

NAVAL1.IDX	PIECE COMMON INFORMATION DATABASE
NAVAL1.DAT	
NAVAL2.IDX	PIECE POINT INFORMATION DATABASE
NAVAL2.DAT	
NAVAL3.IDX	STRIPPING COMMON INFORMATION DATABASE
NAVAL3.DAT	
NAVAL4.IDX	STRIPPING POINT INFORMATION DATABASE
NAVAL4.DAT	
NAVALF.IDX	STRIPPING PIECE INFORMATION DATABASE
NAVALF.DAT	
STANDARD.IDX	MACRO INFORMATION LIBRARY
STANDARD.DAT	

데이터베이스에 저장되는 형태는 Fig. 9와 같다.

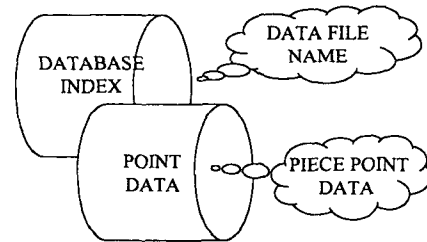


Fig. 9. Types of database

#### 6.1 데이터베이스 디렉토리 관리

데이터베이스는 입·출력 및 데이터의 수정이 많으므로, 사용자가 이용하기 쉽게 Fig. 10과 같이 디렉토리를 구성하였다.

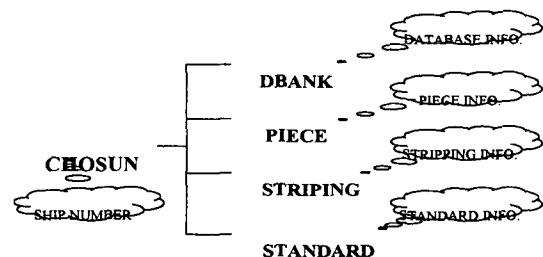


Fig. 10. Example of directory for ship name of CHOSUN

Fig. 10에서 보는 것처럼 호선이 CHOSUN인 선박의 데이터베이스는 DBANK 디렉토리, 부재정보는 PIECE 디렉토리, 스트리핑과 매크로에 관한 정보는 각각 STRIPPING과 STANDARD 디렉토리에 저장하였다.

6.2 부재관리 검증

부재관리에서는 부재 데이터베이스 및 스트리핑 데이터베이스 관리 프로그램으로 구분하였으며, 부재등록의 메인 화면은 Fig. 11과 같다.

Fig. 11과 같이 PIECE D/B에 부재를 등록시킨 후, Fig. 12에서와 같이 스트리핑할 부재를 파일로 생성한다.

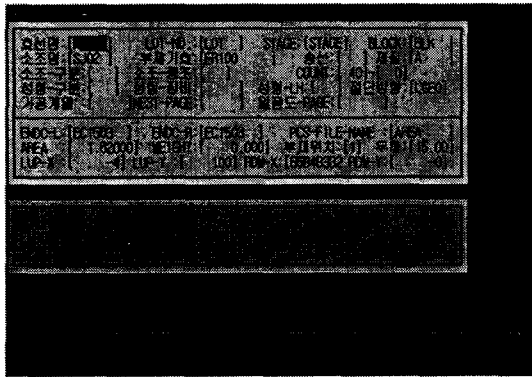


Fig. 11. Main display of PIECE D/B

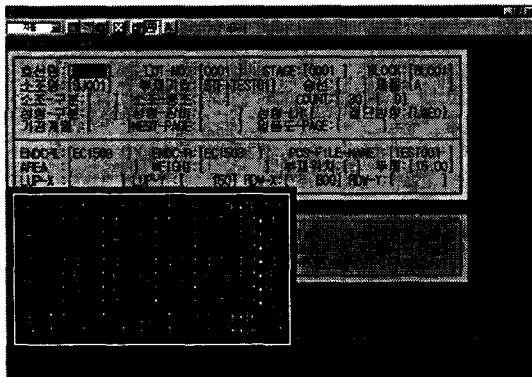


Fig. 12. Example of generating \*.PTC file

스트리핑에 사용할 부재의 생성된 하나의 파일의 내용은 Table. 5와 같다.

Table. 5. Example of \*.PTC file

LINE, 30001,	0.000,	1200.000,	0.000,	100.000,		9,
ARC, 30002,	100.000,	0.000,	0.000,	0.0000000,	1.5707960,	9,
LINE, 30003,	100.000,	0.000,	800.000,	0.000,		9,
LINE, 30004,	800.000,	0.000,	800.000,	80.000,		9,
ARC, 30005,	20.000,	800.000,	100.000,	3.1893311,	4.7123880,	9,
LINE, 30006,	780.023,	99.046,	773.675,	231.905,		9,
LINE, 30007,	773.675,	231.905,	198.390,	1126.794,		9,
LINE, 30008,	198.390,	1126.794,	90.838,	1182.222,		9,
ARC, 30009,	20.000,	100.000,	1200.000,	3.1415920,	4.2365279,	9,
LINE, 30010,	80.000,	1200.000,	0.000,	1200.000,		9,
END						

7. 부재배치전략

생성된 부재들을 이용하여 스트리핑에 사용되는 부재는 스티프너와 같이 어느 정도 폭이 일정하고 외형이 간단한 부재를 선택하였다. 선택조건은 강재의 재질과 두께에 맞게 부재를 선택하게 하였다.

7.1 스트리핑 구현

스트리핑하기 위한 프로그램의 구성은 강재의 재질과 두께에 맞게 부재를 선택하는 "PIECE SELECT"와 선택된 부재를 강재에 배치하는 "AUTORUN"과 배치된 부재를 관리하는 "STRIPPING D/B"로 구분하였다. 프로그램의 흐름도는 Fig. 13과 같다.

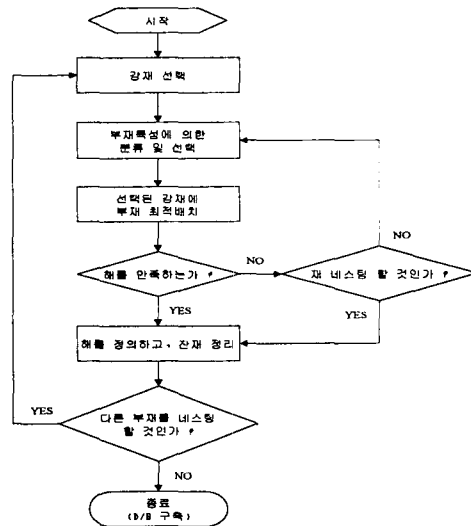


Fig. 13. Flow chart of stripping program

7.2 스트리핑할 부재선택

먼저 강재의 재질과 두께를 입력하면, 강재에 맞는 부재가 선택되고, 강재의 특성에 맞는 부재를 선택하는 순서는 Fig. 14와 같다.

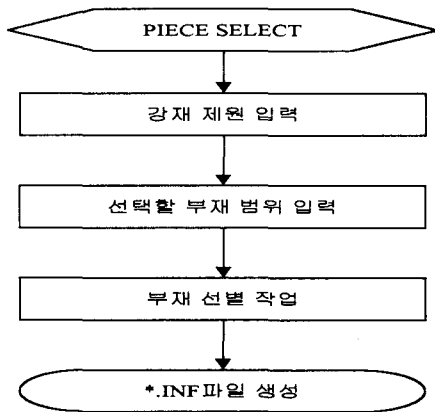


Fig. 14. Flow-chart of piece selection

부재를 선택하기 위하여 먼저 강재의 제원을 입력하게 되는데 강재의 제원과 선택될 부재 분류 범위의 내용은 Table. 6과 같다.

Table. 6. Input values for piece selection

STRIP-PAGE	N2000	RAW-NAME	R1200	RAW-LENGTH	12000			
RAW-BRDTH	3000	RAW-THICK	15.00	RAW-재질	A			
RAW-수량	3	FILE-NAME	N200_01					
OUT-DIR	C:\CHOSUN\stripping\N2000							
FR	호선	1001	LOTNO	1000	STAGE	S1000	BLOCK	F001
TO	호선	1001	LOTNO	1000	STAGE	S1000	BLOCK	F021

Table. 6에서 보는 것처럼 선택할 부재 분류 순서는 호선, LOTNO, STAGE, BLOCK의 순서로 부재를 선별하게 하였다. 스트리핑에 사용할 부재들을 선택하여 하나의 파일인 \*.INF파일로 생성하고, 데이터베이스를 구축한 뒤, 하나의 파일로 생성하고, 강재에 부재를 스트리핑하게 된다.

부재선택을 위하여 입력한 값에 맞는 부재의 목록

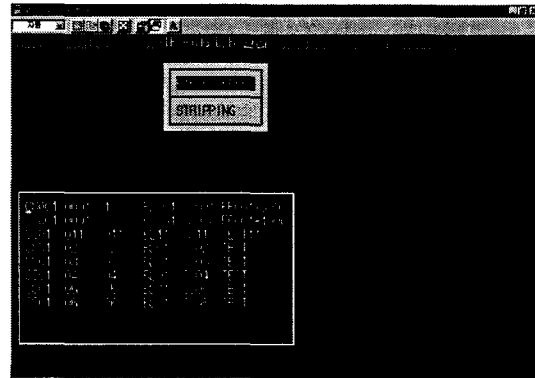


Fig. 15. Selection of piece for stripping

Table. 7. Contents of \*.INF file

구분	내용
TOP	강재의 제원(두께, 재질, 개수, 가로·세로 길이 등)
STRT	선택된 각각 부재들의 제원
LINE, ARC	선택된 부재들의 엔터티
END	선택된 한 부재의 끝
FIN	선택된 부재들의 마지막(EOF)

Table. 8. Example of \*.INF file

```

TOP,stripping ,BONG_HAN, A, 8000.000, 3000.000, 15.000, 5, 1,
STRT,CB ,102 ,102 ,BL102, SJ102, TEST01+12, TEST02, -35.000,200.000,
550.000, -0.000,15.00, 0.000, 40, 1, 102323,
LINE,30001, 50.000, 0.000, 450.000, 0.000,
ARC,30002, 50.000, 500.000, 0.000, 1.5707963, 3.1415927,
LINE,30003, 500.000, 50.000, 550.000, 100.000,
LINE,30004, 550.000, 100.000, 550.000, 200.000,
LINE,30005, 550.000, 200.000, 0.000, 200.000,
LINE,30007, 0.000, 200.000, 0.000, 50.000,
ARC,30006, 50.000, 0.000, 0.000, 0.000, 1.5707963,
END,
STRT,CB ,103 ,103 ,BL103, SJ103, TEST01+13, TEST03, -35.000,200.000,
750.000, -0.000,15.00, 0.000, 40, 1, 130000,
LINE,30001, 0.000, 0.000, 650.000, 0.000,
LINE,30002, 650.000, 0.000, 700.000, 100.000,
LINE,30003, 700.000, 100.000, 750.000, 200.000,
LINE,30004, 750.000, 200.000, 100.000, 200.000,
LINE,30005, 100.000, 200.000, 50.000, 100.000,
LINE,30006, 50.000, 100.000, 0.000, 0.000,
END,
    
```

들을 보여주고, 스트리핑에 사용할 부재를 선택하는 화면은 Fig. 15와 같다.

Fig. 15에서 선택된 부재들을 그룹화하여 하나의 파일인 INF파일로 생성한다. 생성된 INF파일의 내용은 Table. 7과 같고, 생성된 파일의 예는 Table. 8과 같다.

7.3 부재 스트리핑 방법

선택된 부재를 강재에 배치하기 전에 모든 부재들을 폭과 길이 순서로 재 정렬<sup>(3)</sup>하게 된다. 재 정렬을 하는 이유는 스트리핑할 때 먼저 폭의 크기 순서로 배치하고, 다음 부재의 크기 순서로 배치하게 하였다. 부재배치 프로그램의 흐름도는 Fig. 16과 같다.

먼저 강재의 기준점을 정의하면, 처음 부재가 위치하면, 다음 부재의 폭이 결정되어 서로 겹침이 일어나지 않게 배치된다.

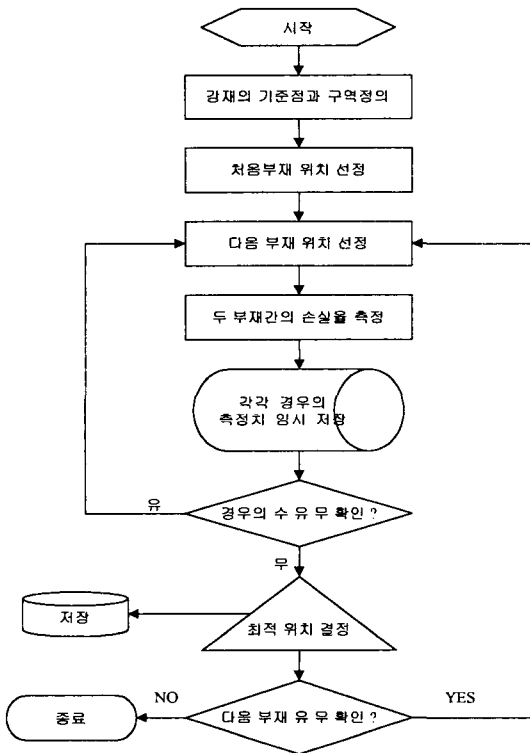


Fig. 16. Flow chart of stripping method

7.4 부재배치 전략

부재배치는 Fig. 17과 같이하고, 두 번째 부재가 놓일 수 있는 모든 경우의 수만큼 두 번째 부재를 놓은 뒤, 각각 경우에 대한 두 부재간의 수율을 측정하고, 강재수율이 가장 좋게 위치한 부재의 상태를 택하게 하였다. 두 부재가 놓일 수 있는 경우의 수는 Fig. 17과 같다.

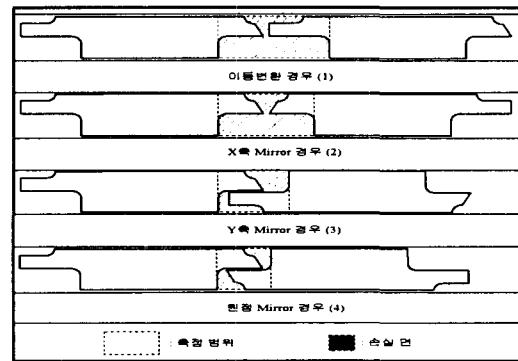


Fig. 17. Cases of setting pieces

7.5 강재 수율측정

Fig. 17과 같이 부재를 배치하고, 강재수율을 측정하는 방법은 Fig. 18과 같이 비교할 두 부재 모두를 Y축 방향으로 등 간격으로 나눈 뒤, 만나는 모든 선들에 대한 좌표값을 구한다. 구해진 좌표값 모두에 대해 한 점을 기준점으로 하고, 동일한 Y축 선상의 좌표와 주어진 거리만큼 이동시킨 후, 다른 점들이 서로 겹치는 점이 없는 좌표를 찾기 위하여 반복한 후, 겹침이 이루어지지 않는 위치의 좌표값을 구하여 저장시키고, 부재가 놓일 수 있는 모든 경우의 수를 반복시켜 부재가 위치할 수 있는 점을 선택<sup>(4), (7)</sup>하게 하였다.

이와 같이 부재가 놓일 수 있는 경우를 모두 배치하였으면, 최적의 위치를 선택해야 하는데 선택하

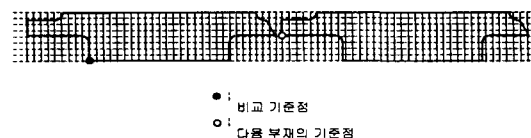


Fig. 18. Setting of position for next piece



는 방법은 두 부재가 놓인 공간의 면적을 Fig. 18과 같이 비교하여 면적이 적게 위치한 부재를 선택하게 하였다. Fig. 17의 경우는 경우(4)를 택하게 된다.

두 번째 부재의 위치를 결정하였으면, 다음부재의 위치는 처음 실행한 방법과 같이하여 다음의 부재 위치를 결정하게 된다.

이러한 방법으로 부재를 놓는 도중 강재의 끝 부분에 부재를 놓을 때는 부재와 강재의 우측면이 교집합을 이루면, 놓았던 부재는 버리고 부재의 길이가 다음으로 적은 부재가 선택되어 해를 구하게 된다. 이러한 방법으로 우측면에 최적의 부재를 선택하고, 다음 라인에 놓을 처음 부재를 선택하여 위에 설명한 방법으로 진행된다.

성은 파라미터를 입력하면 부재가 설계되어 선박부재 생성을 간편화하였으며, 생성된 선박 부재관리의 효율성을 높이기 위하여 데이터베이스를 구축하였다.

- (2) 강재 스트리핑에 필요한 STRIP 프로그램을 개발하여 선박부재 중 스티프너를 택하여 시뮬레이션 결과로부터 강재효율을 기존의 스트리핑 방법 이상으로 향상됨과 효율성을 입증하였다.
- (3) 끝으로 개발된 STRIP 프로그램을 중·소형 조선소에 적용하면, 보편화된 PC에 의한 선박부재 데이터베이스관리와 자동 스트리핑 프로그램에 의하여 강재의 수율을 높일 수 있을 것으로 사료되고, 본 연구와 연계하여 최적 네스팅에 대한 연구가 지속적으로 수행되어야 할 것이다.

8. 검 증

후 기

최종적으로 부재가 배치된 그림은 Fig. 19와 같다.

이 논문은 한국과학재단 지정 지역협력연구센터인 조선대학교 수송기계부품 공장자동화 연구센터의 1998년도 연구비의 지원에 의해 연구되었음.

Fig. 19과 같이 강재에 배치된 부재의 정보는 데이터베이스에 저장되어 확인을 할 수 있게 하였다.

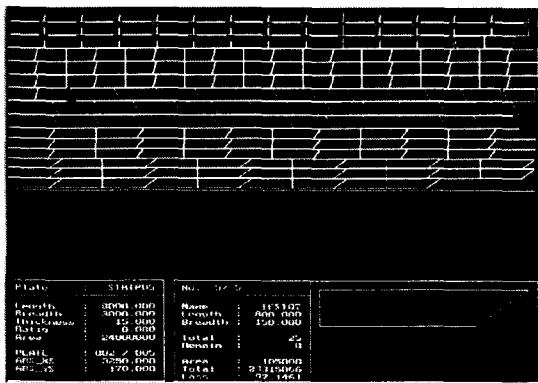


Fig. 19. Case of automatic stripping

9. 결 론

- (1) 개발한 ENDCUT-DEFINE 프로그램에 의하여 부재 외곽형상을 라이브러리화하고, 부재생

참 고 문 헌

1. 김성조, 국형준, "데이터 구조", 도서출판 생능, 1994. pp. 78~95
2. 박제용, 이현상, "PC - CADRA에서 선체 내부재 부재(PIECE) 생성에 관한 연구", 한국해양공학회지, Vol.11 No.3, 1997년 9월. pp. 191~199
3. 박제용, 이현상, 한창봉, "PC-CADRA에서 내부재 피스(Piece) 네스팅에 관한 연구", 한국해양공학회지, Vol.11 No.4, 1997년 11월. pp. 239~248
4. 이진우, "컴퓨터그래픽과 CAD", 영지문화사, 1994. pp. 65~78
5. 이병욱, "데이터 베이스 시스템", 생능출판사, 1995. pp. 112~135
6. 이현상, "선박의 내부재 절단 자동화 시스템 개발에 관한 연구" 공학박사학위논문, 인하대학교,

- 1997.
7. A. Albano and R. Orsini, "A Heuristic Solution of the Rectangular Cutting Stock Problem", The Computer Journal, Vol.23 No.4, Nov. 1980, pp. 338~343.
  8. A. Sasahara, "Contribution to Nesting System: One Approach to the Two-Dimensional Allocation Problem", International Conference on Computer Applications in the Automation of Shipyard Operation and Ship Design, Paper No.VII-5, Tokyo, Aug. 1973, pp. 501~506.
  9. B. Sperling, "Nesting is more than a Layout Problem", International Conference on Computer Applications in the Automation of Shipyard Operation and Ship Design III, Univ. of Strathclyde, June 1979, pp. 287~294.