

파라메트릭 매크로를 이용한 부재 생성 및 네스팅에 관한 연구

한 창 봉* · 박 제 응**
(99년 1월 16일 접수)

A Study on the Generating Stiffener and Nesting Using the Parametric Macro

Chang-Bong Han* · Je-Woong Park**

Key Words : TRIBON(조선전용 CAD S/W), Autorun(자동 스트리핑 프로그램), STRIP(부재를 강재에 배치하는 작업), ENDCUT(부재 모서리 부분), INF 파일(부재의 배치를 위한 선택된 부재들의 정보파일), RPT 파일(강재에 배치된 부재들의 정보파일), PTC 파일(스트립에 사용할 부재 각각의 데이터 파일), RDPTC (스트립에 사용할 PTC파일을 선택하기 위하여 데이터베이스에서 불러들이는 기능)

Abstract

Currently, large size shipyards have performed the generating of piece and the nesting using shipbuilding-oriented CAD program, which requires high skilled operator and costs a lot. So, medium and small size shipyards have pursued other way to obtain the generating and the nesting due to an economic and a technical problems. Therefore, 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.

1. 서 론

현재 국내 조선소에서는 여러 종류의 네스팅 프로그램을 사용하고 있으며 자체적으로 개발을 시도하고 있으나, 주로 고가이고 고급 인력을 필요로

하는 대형 컴퓨터를 기반으로 하고있는 대형조선소의 실정에 맞게 개발을 하고 있다. 또한 국내 대형조선소에서는 TRIBON과 같은 조선 전용 프로그램^{8,9)}을 활용하여 네스팅 작업을 하고 있으나, 이러한 네스팅 프로그램들은 워크스테이션(work-

* 조선대학교 대학원

** 종신회원, 조선대학교 선박해양공학과

station)이나 마이크로 컴퓨터에서 사용되는 것이므로 사용법을 배우는데도 많은 시간이 걸리고, 장비의 고가로 어려운 점이 많아 중·소형 조선소에서 이러한 프로그램들의 사용은 현재까지 엄두도 내지 못하고 있는 실정이다. 그래서 본 연구에서 중·소형 조선소의 조선 기술 향상 및 선박 품질을 한 단계 높이기 위한 목적으로 PC에서 사용가능⁶⁾한 내부재 피스 생성과 스트립너와 브라켓 종류의 네스팅 프로그램 중 스트립(strip) 프로그램을 개발하기에 이르렀다.

2. 부재 생성 프로그램 흐름도

부재 생성 프로그램의 흐름도는 Fig. 1과 같다.

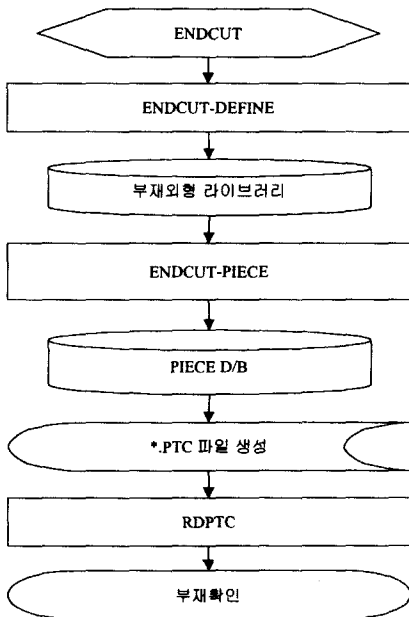


Fig. 1 Flow chart of piece generation

본 연구는 PC(Personal Computer)에서 구동하는 CADRA 프로그램의 매크로를 이용하여 생산도면의 부재들을 생성하는 프로그램으로 도면으로부터 받아들이는 데이터값으로 형상들을 정의한 후 라이브러리를 구축^{1,2)}하였다.

부재 모서리의 형상을 매크로로 정의하기 위해서는 Fig. 2와 같은 형상을 Table 1과 같은 양식으

로 자료를 입력하게 하였다.

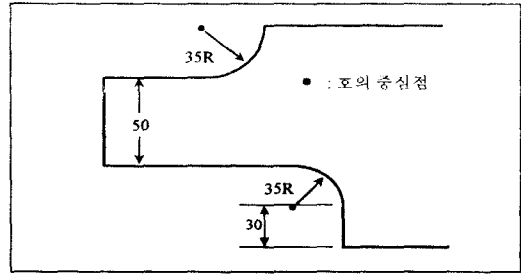


Fig. 2 Form of ENDCUT

Table 1 Parametric characteristic of ENDCUT-DEFINE

ID-NAME	EC1504		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 1에 입력받는 데이터의 내용은 Table 2와 같다.

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

부재를 생성하기 전에 매크로에 정의한 예를 살펴보면, ID-NAME이 EC1504이고, 폭이 150인 부재의형을 정의하는 ENDCUT-DEFINE 프로그램에서 받아들이는 파라미터는 Fig. 3과 같다.

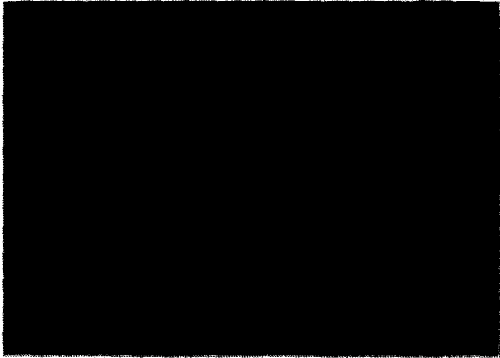


Fig. 3 Example of ENDCUT-DEFINE program

Fig. 3과 같이 정의된 부재의형은 하나의 ID로 라이브러리에 저장되어 부재를 생성하는데 사용된다.

3. 부재생성 방법

정의된 매크로에 변수를 입력하여 부재를 생성하는데 부재를 생성하는 프로그램은 브라켓과 스티프너 생성하는 방법으로 나누어 구성하였다.

1) 브라켓생성 방법

브라켓생성 프로그램의 흐름도는 Fig. 4와 같으며 각 단계별 내용은 다음과 같다.

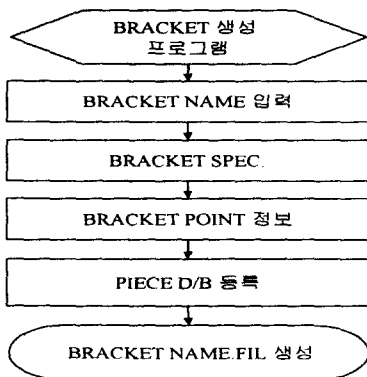


Fig. 4 Flow-chart of bracket generation

브라켓을 생성하기 위하여 입력하는 이름과 내용들은 생산도면에 정의된 Table 3과 같은 데이터를 Fig. 5에서와 같이 정의될 수 있도록 하였다.

Table 3 Parametric characteristic of bracket piece

FILE-NAME	BRK001	호선명	CS001	LOT-NO	100001
STAGE	1001	BLOCK	BL001	소조	SJ001
부재기호	FR001+3000	재질	A	두께	15
수량	20	가로	300	세로	800
사잇각	15	DEPTH	30R	DEPTH	15
CORNER 반경	50	TAIL	50		
세로_X	20	세로_Y	50	세로_R	15
가로_X	50	가로_Y	20	가로_R	15

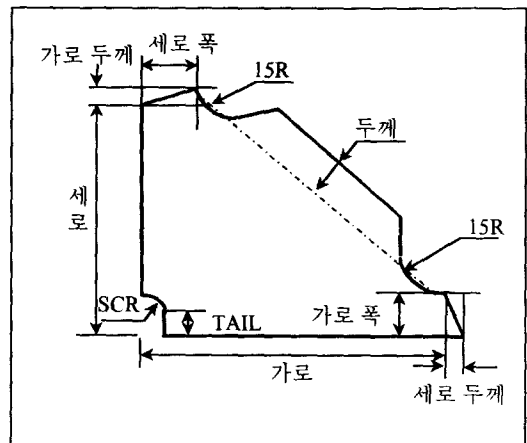


Fig. 5 Form of bracket piece

이와 같이 입력된 정보로 생성된 부재는 네스팅을 위하여 부재 데이터베이스⁵⁾에 등록하였다.

2) 스티프너생성 방법

스티프너생성 프로그램의 흐름도는 Fig. 6과 같다. 스티프너를 생성하기 위하여 입력하는 이름과 내용은 Table 4와 같고, 입력받은 값은 Fig. 7에서 보는 것과 같이 스티프너가 형성되게 하였다.

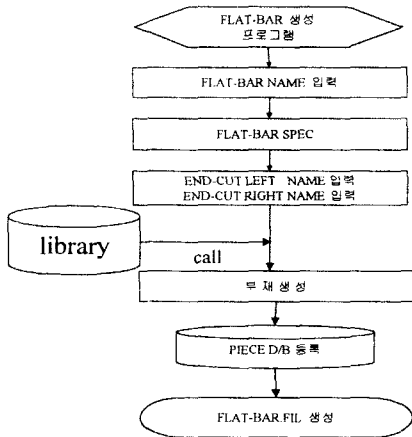


Fig. 6 Flow-chart of ENDCUT

Table 4 Parametric characteristic of stiffener

FILE-NAME	STF001	호선명	CS001	LOT-NO	10001
BLOCK	BL001	소조명	SJ001	재질	A
수량	20	두께	15	부재기호	FR200+ 3000
LENGTH	2400			WEB높이	150
END-CUT LEFT NAME	EC1504		END-CUT RIGHT NAME	EC1501	

3. 생성된 부재의 확인

1) 스티프너 생성

정의된 형상을 이용하여 부재를 생성하게 되는데 정보를 받아들이는 화면은 Fig. 7과 같다.

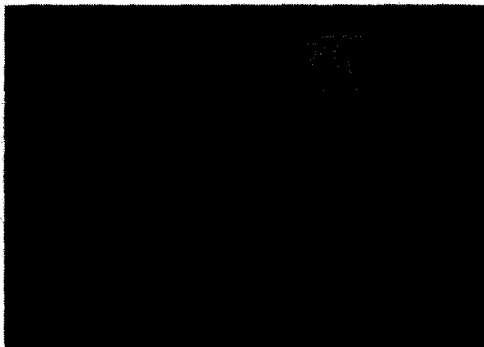


Fig. 7 Example of ENDCUT-PIECE program

Fig. 7과 같이 생성된 부재는 부재 데이터베이스에 등록되고, 스트리핑에 사용할 수 있도록 STF04.PTC파일을 생성한다.

생성된 부재를 PC에서 운용되는 CADRA 화면에서 확인한 장면은 Fig. 8과 같다.

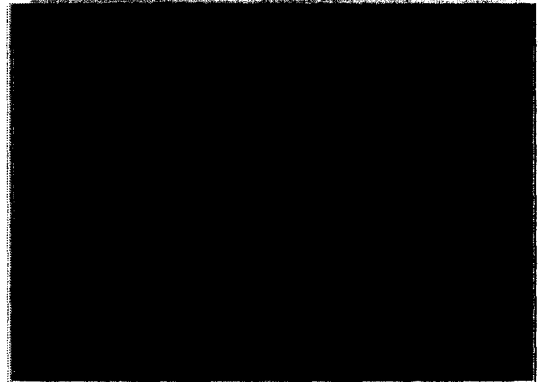


Fig. 8 Identification of generating piece

2) 브라켓 부재 생성

브라켓 생성 프로그램에서 받아들이는 파라미터는 Fig. 9와 같다.



Fig. 9 Example of input data for bracket.

Fig. 9의 파라미터를 입력받아 생성된 브라켓은 Fig. 10과 같다.

4. 부재관리

데이터베이스는 크게 두 부분으로 나누어 구축

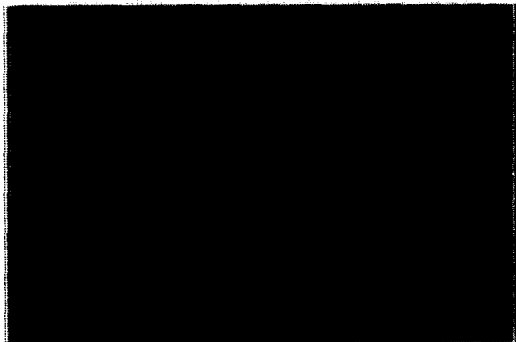


Fig. 10 Example of generating bracket.

되었다. 먼저, 책의 목차부분과 비슷한 인덱스 파일인 *.IDX파일과, 책의 내용에 비교할 수 있는 데이터 파일인 *.DAT파일로 구분하였다. 데이터베이스의 이름이 "NAVAL"이라면 인덱스 파일과 데이터 파일의 이름은 Table 5와 같다.

Table 5 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 형태를 그림으로 그리면 Fig. 11과 같다.

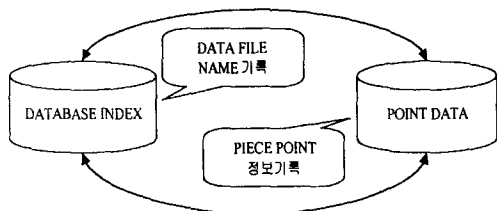


Fig. 11 Types of database

1) 데이터베이스 디렉토리 관리

데이터베이스에 대한 입·출력 및 데이터의 수정이 많으므로, 사용자가 알아보기 쉽고, 이용하기 쉽게 디렉토리를 구성하였다.

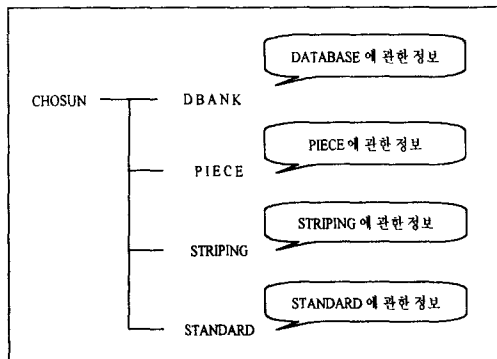


Fig. 12 Example of directory for ship name of CHOSUN

Fig. 12에서 보는 것처럼 호선이 "CHOSUN"인 선박의 데이터베이스는 "DBANK" 디렉토리, 부재 정보는 "PIECE" 디렉토리, 스트리핑과 매크로에 관한 정보는 각각 "STRIPPING"과 "STANDARD" 디렉토리에 저장하였다.

2) 부재 관리 검증

데이터베이스 관리에서는 부재 데이터베이스 및 스트리핑 데이터베이스 관리 프로그램으로 부재등록의 메인 화면은 Fig. 13과 같다.

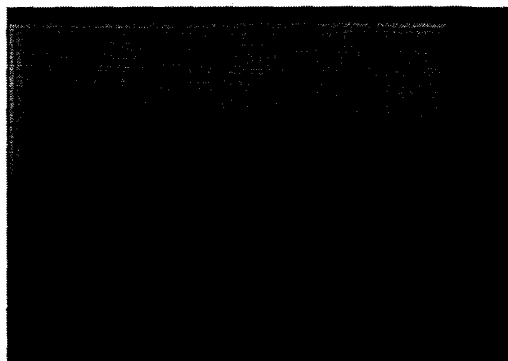


Fig. 13 Main display of PIECE D/B

Fig. 13과 같이 PIECE D/B에 부재를 등록시킨 후, 스트리핑에 사용할 부재인 *.PTC 파일의 정의와 생성단계는 Fig. 14와 같다.



Fig. 14 Example of generating *.PTC file

스트리핑에 사용할 *.PTC파일의 내용은 Table 6과 같다.

Table 6 Example of *.PTC file

LINE, 30001, 0.000, 1200.000, 0.000, 100.000,	9,
ARC, 30002, 100.000, 0.000, 0.000, 0.00000000, 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	

5. 부재배치 전략

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

1) 스트리핑 구현

스트리핑하기 위한 프로그램의 구성은 강재의 재

질과 두께에 맞게 부재를 선택하는 "PIECE SELECT"와 선택된 부재를 강재에 배치하는 "AUTORUN"과 배치된 부재를 관리하는 "STRIPPING D/B"로 구분하였다.

프로그램의 흐름도는 Fig. 15와 같다.

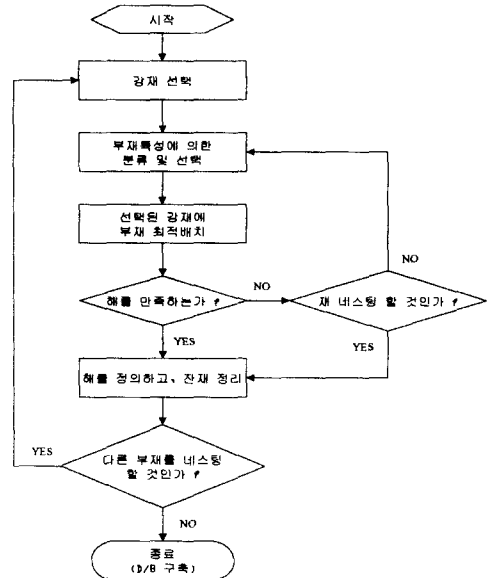


Fig. 15 Flow-chart of stripping program

1) 스트리핑할 부재선택

먼저 강재의 재질과 두께를 선택하고, 그 강재에 맞는 부재를 선택하게 된다. 강재의 특성에 맞는 부재를 선택하는 순서는 Fig. 16과 같다.

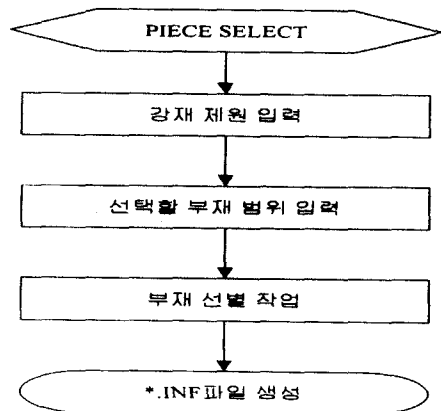


Fig. 16 Flow-chart of piece selection

부재를 선택하기 위하여 먼저 강재의 재원을 입력하게 되는데 강재의 재원과 선택될 부재 분류 범위의 내용은 Table 7과 같다.

Table 7 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 8에서 보는 것처럼 선택할 부재 분류 순서는 "호선, LOTNO, STAGE, BLOCK"의 순서로 부재를 선별하게 하였다. 스트리핑에 사용할 부재들을 선택하여 하나의 파일인 *.INF파일로 생성하고, 데이터베이스를 구축한 뒤, 하나의 파일로 그룹화한 뒤, 강재에 부재를 스트리핑하게 된다.

부재선택을 위하여 입력한 값에 맞는 부재의 목록들에서 스트리핑에 사용할 부재를 선택하는 화면은 Fig. 17과 같다.

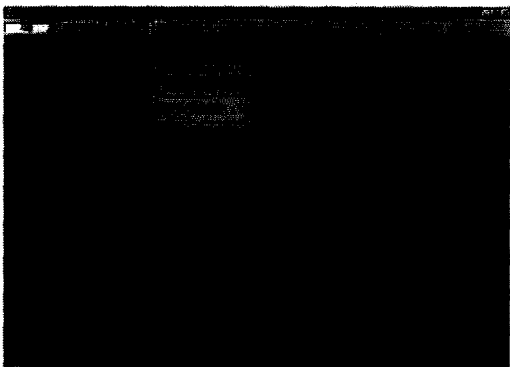


Fig. 17 Selection of piece for stripping

Fig. 17에서 선택된 부재들을 하나의 파일인

INF파일로 그룹화되어 생성된다. 생성된 INF파일의 내용은 Table 8과 같고 생성된 파일의 예는 Table 9와 같다.

Table 8 Contents of *.INF file

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

Table 9 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,
STRT,CB ,104 ,104 ,BL104, SJ104,TEST01+14 ,TEST04, -35.000, 200.000,
500.000, -0.000,15.00, 0.000, 40, 1, 60000,
LINE,30001, 200.000, 0.000, 300.000, 0.000,
LINE,30002, 300.000, 0.000, 450.000, 50.000,
LINE,30003, 450.000, 50.000, 500.000, 100.000,
LINE,30004, 500.000, 100.000, 350.000, 150.000,
LINE,30005, 350.000, 150.000, 300.000, 200.000,
LINE,30006, 300.000, 200.000, 200.000, 200.000,
    
```

2) 부재 스트리핑 방법

선택된 부재를 강재에 배치하기 전에 모든 부재들을 폭과 길이 순서로 재 정렬⁽³⁾하게 된다. 재 정렬을 하는 이유는 스트리핑할 때 먼저 폭의 크기 순서로 배치를 하고, 다음 길이의 크기 순서로 부재를 배치하게 하였다. 부재를 자동으로 배치하는 프로그램의 흐름도는 Fig. 18과 같다.

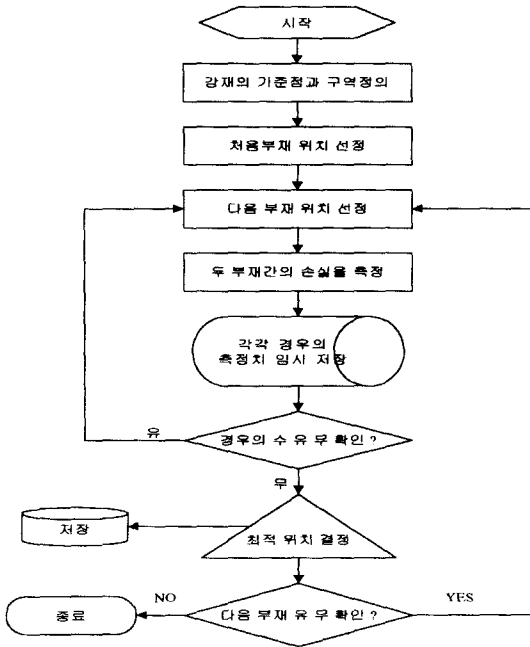


Fig. 18 Flow-chart of stripping method

먼저 강재의 기준점과 구역을 정의하면, 처음 부재가 자리를 찾고, 다음 부재들의 폭이 결정되어 서로 겹침이 일어나지 않게 배치된다.

3) 부재배치 전략

부재배치는 Fig. 19와 같이하고, 두 번째 부재가 놓일 수 있는 모든 경우의 수만큼 두 번째 부재를 놓은 뒤, 각각 경우에 대한 두 부재간의 수율을 측정된 뒤, 강재수율이 가장 좋게 위치한 부재의 상태를 택하게 하였다. 두 부재가 놓일 수 있는 경우의 수는 Fig. 19와 같다.

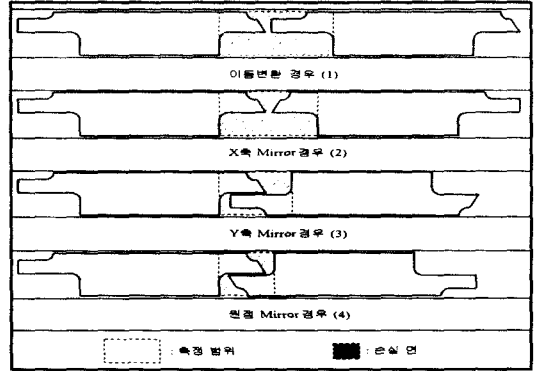


Fig. 19 Cases of setting pieces

4) 강재 수율측정

Fig. 19와 같이 부재를 배치하고, 강재수율을 측정하는 방법은 Fig. 20과 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^{(4), (7)}하게 하였다.

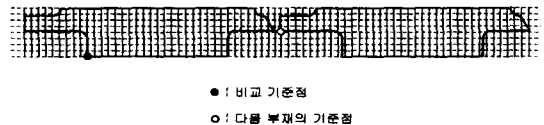


Fig. 20 Setting of position for next piece

이와 같이 부재가 놓일 수 있는 경우를 모두 배치하였으면, 최적의 위치를 선택해야 하는데 선택하는 방법은 두 부재가 놓인 공간의 면적을 Fig. 20과 같이 비교하여 면적이 적게 위치한 부재를 선택하게 하였다. Fig. 19의 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.

5. 검 증

최종적으로 부재가 배치된 그림은 Fig. 21과 같고, Tribon에서 배치된 그림은 Fig. 22이다. 두 프로그램간의 비교에서 강재수율에는 차이가 없었다.

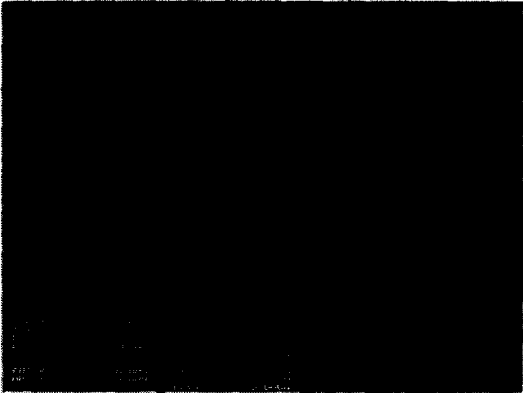


Fig. 21 Case of automatic stripping

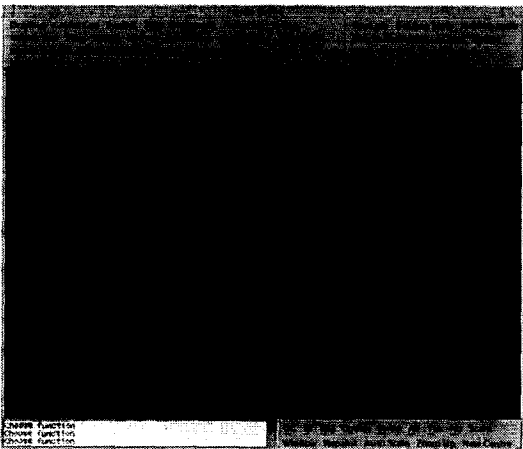


Fig. 21과 같이 강재에 배치된 부재의 정보는 데

이터베이스에 저장되어 확인을 할 수 있게 하였다.

6. 결 론

- 1) 개발한 ENDCUT-DEFINE 프로그램에 의하여 부재 외곽형상을 라이브러리화하고, 부재생성은 파라메터를 입력하면 부재가 설계되도록 하여 선박부재 생성을 간편화하였으며, 생성된 선박부재관리의 효율성을 높이기 위하여 데이터베이스를 구축하였다.
- 2) 강재 스트리핑에 필요한 STRIP 프로그램을 개발하여 선각부재 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.

후 기

이 논문은 1997년도 조선대학교 학술연구비 지원에 의하여 연구되었음.

참고문헌

- 1) 김성조, 국형준, "데이터 구조", 도서출판 생능, 1994
- 2) 박제응, 이현상, "PC - CADRA에서 선체 내부재 부재(PIECE) 생성에 관한 연구", 한국해양공학회지, Vol.11 No.3, 1997. 9
- 3) 박제응, 이현상, 한창봉, "PC-CADRA에서 내부재 피스(Piece) 네스팅에 관한 연구", 한국해양공학회지, Vol.11 No.4, 1997. 11
- 4) 이진우, "컴퓨터그래픽과 CAD", 영지문화사, 1994
- 5) 이병욱, "데이터 베이스 시스템", 생능출판사,

1995

- 6) 이현상, "선박의 내부재 절단 자동화 시스템 개발에 관한 연구" 공학박사학위논문, 인하대학교, 1997
- 7) Albano, A. and R. Orsini, "A Heuristic Solution of the Rectangular Cutting Stock Problem", The Computer Journal, Vol.23 No.4, pp.338-343, Nov. 1980
- 8) Sasahara, A., "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, pp.501-506, Aug. 1973
- 9) Sperling, B., "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, pp.287-294, June 1979