

공학석사 학위논문

유전자 알고리즘을 이용한 조선 소조립 로봇용접  
공정 최적화 및 3D 시뮬레이션을 통한 검증

**Optimization of Robot Welding Process  
using Genetic Algorithm and Verification by 3D Simulation  
in Subassembly of Shipyard**

지도교수 박 주 용

2005년 2월

한국해양대학교 대학원

해양시스템공학과

강 현 진

# 목 차

<b>List of tables</b> .....	<b>III</b>
<b>List of figures</b> .....	<b>IV</b>
<b>Abstract</b> .....	<b>VI</b>
1. 서론 .....	1
1.1 연구 배경 및 필요성 .....	1
1.2 연구 내용 .....	2
2. 조선 소조립 로봇 용접 공정 .....	3
2.1 소조립 공정 .....	3
2.2 로봇 용접 공정 .....	4
2.3 정보 모델링 .....	5
2.4 소조립 로봇 용접 공정의 현재 .....	6
2.5 소조립 로봇 용접 공정의 문제 제기 .....	7
3. 작업 계획 .....	9
4. 작업량의 배분 .....	12
4.1 작업량 배분의 필요성 .....	12
4.2 작업량 배분 시 고려할 요소 및 목적 함수 .....	14
4.3 작업량 배분 방법 .....	15
5. 작업 순서의 결정 .....	17
5.1 작업 순서의 중요성 .....	17
5.2 작업 순서 결정 문제의 특성 .....	17

5.3	작업 순서 결정 방법의 모색	18
5.4	유전자 알고리즘	18
5.4.1	유전자 알고리즘의 개요	18
5.4.2	유전자 알고리즘의 요소	19
5.4.3	유전자 알고리즘의 절차	22
5.4.4	스키마 정리	23
5.5	유전자 알고리즘의 적용	26
5.5.1	표현형의 설정	26
5.2.2	초기모집단	27
5.2.3	적응도 평가	27
5.2.4	선별	27
5.2.5	교차	28
5.2.6	돌연변이	29
5.2.7	종료 조건	29
6.	용접 방향의 결정	30
7.	작업계획을 고려한 시뮬레이션	31
7.1	시뮬레이션이란 무엇인가	31
7.2	3차원 시뮬레이션의 선택	31
7.3	로봇 용접 공정의 3차원 시뮬레이션	32
7.3.1	로봇 용접 공정 시뮬레이터(Planning)	34
7.3.2	로봇 용접 공정 시뮬레이터(Motioning)	37
7.4	작업 계획의 적용	38
7.5	시뮬레이션 조건	38
7.6	시뮬레이션 결과	39
8.	결론	42
	참고 문헌	43

## List of tables

Table 4.1 Cycle time for work area divided in 3 geometry .....	29
Table 5.1 Selection probability by roulette wheel .....	28
Table 5.2 Adjacent gene .....	28
Table 7.1 Robot specification .....	32
Table 7.2 Robot position .....	33
Table 7.3 Comparison of cycle time for distribution .....	40
Table 7.4 Comparison of cycle time according to change of working sequence using GA .....	41

## List of figures

Fig. 2.1 Layout of subassembly line .....	3
Fig. 2.2 Sub-process of subassembly .....	3
Fig. 2.3 Robot welding shop and gantry-robot .....	4
Fig. 2.4 Sequence diagram of robot welding process .....	6
Fig. 2.5 Skid pattern .....	7
Fig. 3.1 Classes of program .....	9
Fig. 3.2 Product data .....	10
Fig. 3.3 Flow chart of planning .....	11
Fig. 4.1 Arrangement of product .....	12
Fig. 4.2 Work area .....	13
Fig. 4.3 Cycle time of 3 geo and even distribution .....	13
Fig. 4.4 Welding direction .....	14
Fig. 4.5 Concept of distribution .....	15
Fig. 5.1 Feature and solution of TSP .....	17
Fig. 5.2 General structure of genetic algorithm .....	19
Fig. 5.3 Single point crossover and two point crossover .....	21
Fig. 5.4 Mutation .....	21
Fig. 5.5 Procedure of Genetic Algorithm .....	22
Fig. 5.6 The concept of application .....	26
Fig. 5.7 The string of solution .....	27
Fig 5.8 Improvement of solution by generation .....	29
Fig 6.1 Concept of algorithm for welding direction .....	30

Fig 7.1 Welding process .....	33
Fig 7.2 Application of robot position .....	33
Fig 7.3 Real shop and virtual shop .....	33
Fig 7.4 Floor and flat bar .....	34
Fig 7.5 Arrangement of products .....	35
Fig 7.6 Determination of the work area .....	35
Fig 7.7 Procedure of simulation .....	36
Fig 7.8 Output file of planning .....	37
Fig 7.9 Application of planning .....	38
Fig 7.10 Schedule chart .....	38
Fig 7.11 BOM(Bills Of Material) .....	39
Fig 7.12 Graph of difference for working time .....	40

**Optimization of Robot Welding Process  
using Genetic Algorithm and Verification by 3D Simulation  
in Subassembly of Shipyard**

*Hyun-Jin, Kang*

**Division of Ocean Systems Engineering**

**Graduate School of Korea Maritime University**

**Abstract**

This research is carried out to improve productivity of robot welding shop in subassembly line. This is a part of full automation of robot welding shop and it is realized as the 3D simulator for robot welding. UML and IDEF methods are used to analyze system to be model and the analysis is to be the foundation of simulator. Robot welding shop is a semi-mechanized shop. So this shop has possibility to be mechanization and it can be simulated precisely than other shop operated by a worker.

But there are some problems to be considered. It is caused by many machines, non-fixed materials. The problems are determination of working sequence and work area and these need other optimum techniques. The concept of TSP and genetic algorithm are used to find shortest work sequence. Simulator for robot welding shop is implemented to verify the result of optimum plans and to estimates cycle time for specific amount of work be done.

# 제 1 장 서 론

## 1.1 연구 배경 및 필요성

선체는 수많은 외판 및 내부재들을 서로 용접하는 과정을 통해 형성된다. 조선에 있어 용접은 큰 비중을 차지하며 용접 공정에서 로봇을 이용하기도 한다. 조선산업의 자동화의 진행은 다른 산업의 자동화보다 더디며 이는 그 대상 작업이 어렵기 때문이다. 즉, 제품의 규모가 크며, 확실성 및 규칙성이 떨어지기 때문이다.

자동화는 세부 준비 작업을 상당히 많이 필요로 한다. 실제 작업을 하는 것은 자동화시스템에서 전체의 40%밖에 차지하지 못한다. 따라서 작업량을 기준으로 한 생산성이 자동화 시스템이 아닌 경우에 비해 크게 높은 편은 아니다. 하지만 자동화로 인한 부수적인 효과 등으로 인해 자동화는 계속 늘어날 것이다. 소조립 공정에서의 자동화도 마찬가지이다. 그러나 소조립 공정 역시 자동화를 위해서는 앞서 말한 요인들로 인해 해결해야 할 몇 가지 문제가 있다. 조선업에서 소조립 공정의 자동화는 대부분의 조선소에서 해결과제로 남아 있는 상태이다.

조립공정은 크게 소조립과 조립으로 나뉘며, 소조립과 조립공정의 특성상 로봇이 공정에 이용되는 방식이 다르다. 그 중 소조립 공정에서 로봇의 이용은 그 적용에 있어 제약이 있는데, 그것은 매 공정에서 들어오는 부재들의 위치가 고정되지 않는다는 점과 여러 로봇이 상호적으로 움직여야 한다는 것이다. 반면 조립공정은 부재의 위치가 정해지므로 설계정보만으로 충분히 세부 부재 위치(seam)의 파악이 가능하며 로봇이 움직이는 구역 및 용접부가 다소 고정적이다.

소조립 공정은 대부분의 조선소에서 공정의 일부를 로봇으로 대체하고 있으며, 이를 위해 연구하고 있다. 소조립 공정의 로봇 시스템은 대개 겐트리와 로봇의 혼합 시스템을 사용하고 있는데 그 방식에 있어서는 다소 차이가 있다. 먼저 로봇의 수에 관해서 말하자면, 하나의 로봇으로 작업함으로써 여러 로봇간의 충돌 문제를 미연에 방지하도록 하기도 하며, 동시에 많은 물량을 처리하기 위해 여러 기의 로봇을 구동하기도 한다. 또한 겐트리에 로봇이 부착되는 방식도 여러 가지 인데 겐트리에 하나의 로봇이 한 면씩 용접하는 경우도 있고, 두 기 로봇을 대칭으로 부착하여 양면을 동시에 용접할 수 있게 하기도 한다. 후자의 경우 선체가 좌우대칭이라는 점과 양쪽 seam이 같은 형상이라는 점에 근거하여 양면 필렛(fillet) 용접의 경우 두 기의 로봇이 각각 한 면씩 작업을 담당하도록 하기 위함이었다. 그러나 실제로 선체는 좌우대칭이 되는 부분이 그리 많지 않으며, 생산에 있어 양면의 작업방식(seam의 길이 및 로봇 자세)이 항상 일치하는 것이 아니어서 논의의 여지가 있다. 이렇게 자동화에 있어서 겐트리 및 로봇을 이용하는 방식에 따라 그 어려움의 정도가 달라진다.



## 1.2 연구 내용

조선의 소조립 공정 중 로봇용접공정을 분석하고 3차원 공간에 실제 공장을 대신할 수 있는 virtual shop을 구현하였으며 작업 계획 및 작업 시간 예측을 위한 시뮬레이터를 구현하였다. 제 1장 서론에 이어 제 2장에서는 조선 소조립 로봇용접 공정에 대하여 기술하였고 제 3장에서는 첫 번째 작업 계획인 작업 물량을 배분하는 방법에 대해서 논의하였다. 제 4장은 두 번째 작업 계획인 작업 순서를 결정하는 방법과 이에 적용된 알고리즘인 유전자 알고리즘에 관해, 그리고 제 5장은 용접 순서를 결정하는 알고리즘에 관해 기술하였다. 제 6장은 시뮬레이션에 관한 내용으로 이것은 작업계획 결과를 적용한 시뮬레이션이다.

## 제 2 장 조선 소조립 로봇 용접 공정

### 2.1 소조립 공정

소조립 공정의 선체의 내판 및 외판의 보강재가 될 부재를 용접하는 공정이다. 본 연구에서 대상으로 하는 소조립 공정은 Tact Time 생산방식에 따라 운용되는 공정으로, 작업자의 위치는 고정되어 있고 작업물이 Tact Time의 주기를 갖고 공장을 따라 흐른다. 즉, 소조립 공정의 물량은 ‘배재-취부-로봇용접-수동용접-마무리’의 세부공정을 Tact Time에 따라 순차적으로 거친다. Tact Time 생산방식은 흐름생산방식으로 전공정이 동시에 시작하고 동시에 끝나는 것으로, 공정 간의 생산시간을 균형화시키고 공정별 생산능력을 측정 개선하는데 효과적인 방식이다. 생산지시 정보는 단위 Tact Time 생산이 완료된 시점에서 하나의 정보가 주어지고, Tact Time에 맞추어 작업 배분을 하면 애로공정(Bottle Neck)이 바로 눈으로 보이게 되며 각 공정간, 설비간의 문제점을 쉽게 찾아낼 수 있다.

소조립 공정은 여러 라인으로 구성되어 있으며, 각 라인은 물량의 형상에 따라 특징지어진다. Fig. 2.1은 소조립 공장 중 두 개 라인의 Layout을 보여주며 공정은 그림의 오른쪽에서 왼쪽으로 흐른다.

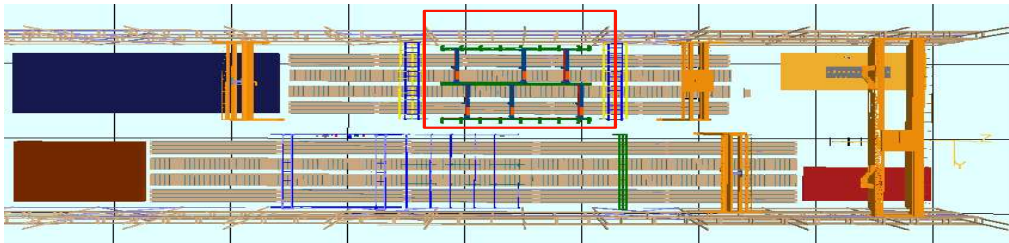


Fig. 2.1 Layout of subassembly line

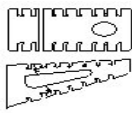
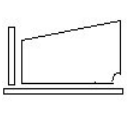
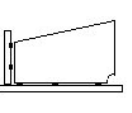
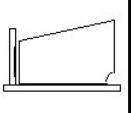
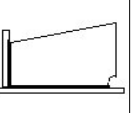
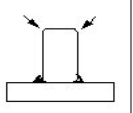
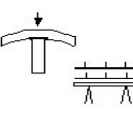
SAW	ARR	FIT	WELD1	WELD2	FINISH	BACK HEATING
						
	Arrangement	Fit-Up	Robot Welding	Manual Welding	Grinding	Modifying

Fig. 2.2 Sub-process of subassembly

Fig. 2.2는 소조립 공정의 세부 공정을 나타낸다. SAW는 주판을 용접하는 공정을 말하며, 다음 단계인 ARR에서는 작업자가 크레인을 조작하여 주판 위에 소부재들을 배치한다. FIT단계에서는 주판의 정확한 위치에 소부재들을 가접하여 세우며, WELD1 공정에서는 본용접을 수행한다. WELD2에서는 WELD1의 작업을 보충하거나 수정하며 FINISH 공정에서는 Grinding을 비롯한 마무리 작업을 한다. 소조립 공정은 이러한 세부 공정의 순차적인 수행으로 구성되며, 여러 소조립 라인은 모두 이러한 구성을 갖추고 있다.

## 2.2 로봇 용접 공정

세부 공정 중 WELD1공정이 로봇에 의해 수행되는 공정의 라인은 직선의 seam을 갖는 workpiece를 용접하는 라인이다. 이 공정을 ‘로봇 용접 공정’이라 부르고 로봇 용접 작업장을 로봇용접장(Robot Welding Shop)이라 칭한다. 로봇용접장은 작업자가 아닌 로봇에 의해 용접이 이루어지며, Fig. 2.1과 Fig. 2.2에서 굵은 사각형으로 표시하였고 다음과 같은 특징이 있다.

<설비 측면>

- ① 로봇 용접장은 두 라인으로 구성된다.
- ② 컨베이어 시스템에 의해 구동된다.
- ③ 하나의 라인에는 3기의 겐트리가 있으며 각각의 Gantry에는 2기의 용접로봇이 대칭으로 부착되어 있다. (Fig. 2.3)
- ④ 두 로봇 사이에는 레이저 다이오드가 있으며 seam의 위치 인식에 사용된다.
- ⑤ 겐트리는 Tx, Ty, Tz, Rz ( 3축 병진, 1축 회전), 로봇은 Tx, Ty, Tz, Rx, Ry, Rz (3축 병진, 3축 회전) 이다.
- ⑥ 모든 ‘겐트리+용접 로봇’(이하 ‘gantry-robot’)은 동일한 작업을 처리 할 수 있는 능력이 있다.



Fig. 2.3 Robot Welding Shop and gantry-robot

<공정 측면>

- ① 두 라인은 별도로 구동된다.
- ② 로봇용접장에는 전단계 공정인 FIT 공정으로부터 취부된 workpiece가 흘러 들어온다.
- ③ 겐트리는 seam의 위치까지 이동하는 역할을 하며 로봇은 실제 용접을 담당한다.
- ④ 작업자는 Teaching Pendant를 이용하여 로봇을 이동시키면서 용접의 시작점과 끝점을 인식시킨다.

## 2.3 정보 모델링

위의 개괄적인 설비 및 공정에 대한 정보는 이어질 작업 계획 설정이나 시뮬레이션에 위한 정보가 되기에는 부족하다. 시뮬레이션을 위한 모델링을 위해서는 공정을 정확히 반영하기 위한 면밀한 정보를 필요로 한다. 어떠한 정보는 필요로 하기도 하고, 어떤 것은 필요 없기도 한다. 이러한 것들을 두서없이 수집하게 되면 제대로 활용하기조차 힘들다. 따라서 체계적이고 절차적인 대상 분석을 위해 두가지 방법론을 적용하였으며, 그것은 UML(Unified Modeling Language)과 IDEF(Integration DEFINition)이다.

### 2.3.1 UML

UML은 시스템을 여러 가지 관점(View)에서 점검하고 관찰할 수 있도록 하는 도구이다. 모든 다이어그램이 모든 UML모델에 등장하는 것은 아니다.

로봇용접장을 관찰하기 위해 사용된 모델은 Class Diagram, Sequence Diagram이다.

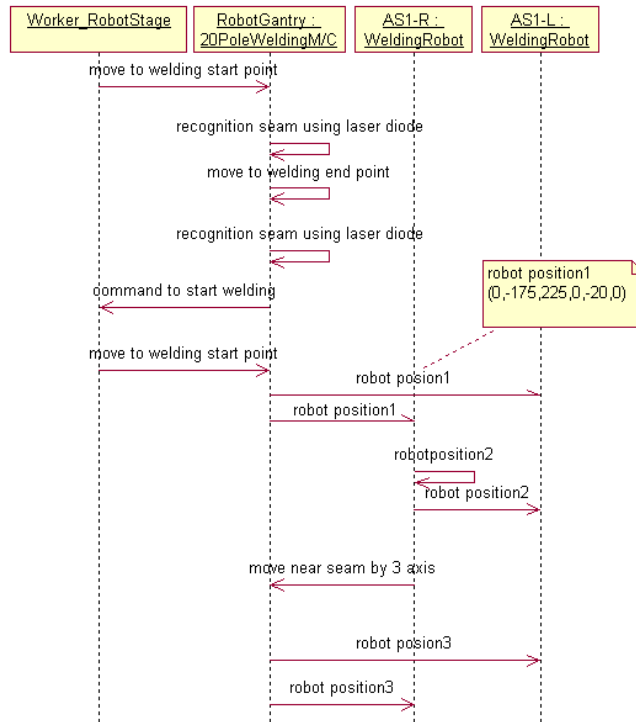


Fig. 2.4 Sequence Diagram of robot welding process

Fig. 2.4는 로봇 용접 공정의 Sequence Diagram이다. 이것은 로봇 용접 공정이 시간이 흐름에 따라 어떻게 흘러가는지 보여준다. 각각의 설비 및 작업자를 개체로 하여 그들 간의 상호 작용을 메시지로 표시한 모델이다.

## 2.4 소조립 로봇 용접 공정의 현재

로봇 용접 공정은 한 명의 작업자가 로봇을 조작하고 있다. 여기서 조작이라는 말은 Teaching Pendant를 사용하여 용접 시 로봇이 제품 내 seam의 위치를 인식할 수 있도록 모든 seam의 용접의 시작점과 끝점을 포인팅하고 간단한 프로그래밍을 하는 것을 말한다. 이 작업자는 하나의 로봇에 대한 티칭 작업이 끝나면 이 로봇에게 용접을 수행하라고 명령하고 다음 로봇을 티칭한다. 따라서 작업자에 의해 로봇에 작업량이 할당되며 작업 순서 및 방향이 결정된다. 로봇용접장에 물량이 들어오면 자동화 시스템에 의해 설계정보로부터 생산정보를 생성하고 작업 계획을 생성하며 설비를 제어하는 일들이 일괄적으로 이루어지지 못하고 있는 실정이다. 따라서 작업 시간이 많이 걸리는 것은 물론이고, 로봇 시스템을 효율적으로

이용하지 못하고 있다는 부담이 있다.

## 2.5 소조립 로봇 용접 공정의 문제 제기

로봇용접장은 다음과 같은 요인으로 인해 자동화에 어려움이 있다.

- ① 제품의 설계정보와 생산정보에의 접근
- ② 제품 형상의 다양함
- ③ 제품 위치 파악의 어려움

스키드(Skid)는 1회에 처리되는 물량을 나타낸다. 스키드 패턴(Skid Pattern)은 작업장에 놓은 작업물의 배치를 말한다. 이것은 작업장내의 작업물의 위치와 그 리스트들을 포함한다.

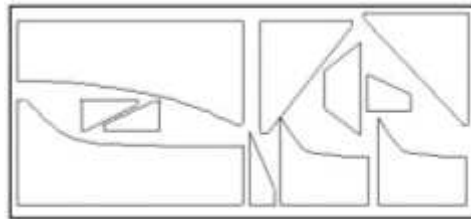


Fig. 2.5 Skid Pattern

ARR 공정에서 이러한 스키드 패턴에 따른 부재의 배치가 이루어지지 않고 작업자의 경험에 따라 배치가 이루어진다. 만약 스키드 패턴이 설계단계에서 만들어진다면 작업자는 이를 따라 부재를 배치할 수 있으며, 위에 제시된 문제들에 보다 쉽게 접근할 수 있다. 그러나 현재 스키드 패턴은 준비상태에 있다.

조립공정은 커다란 블록 하나를 대상으로 작업을 수행하기 때문에 초기에 블록의 위치를 알면 설계 정보를 통해 세부 부재의 위치, 즉 seam의 위치를 알 수 있기 때문에 작업물의 위치를 파악해야 하는 문제는 없다. 반면 소조립 공정에서는 일정하지 않은 제품의 개수와 형상으로 인해 작업장 내에서의 위치를 파악하는 것부터가 쉽지 않다.

또한 소조립 공정의 자동화를 위해 필요한 사항들은 다음과 같다.

- ① 제품의 설계정보와 생산정보에의 접근
- ② 제품의 위치 파악 인식 시스템

- ③ 공정을 수행할 기계간의 작업량 배분 및 각 기계의 작업 수행계획
- ④ 다수의 기계(welding robots, gantry)의 제어
- ⑤ Off-line programming

이러한 요소들 중 작업 수행 계획에 대하여 자세히 알아본다.

### 제 3 장 작업 계획

본 연구에서 작업 계획이라 함은 현 소조립 공정처럼 정형화되지 않은 공정을 처리하기 위한 사전 작업을 말한다. 임의의 수량의 제품과 제품의 위치, 또한 겐트리 및 로봇에 대한 작업량 할당 등의 고려가 없었기 때문에 이러한 작업이 선행되어야 한다.

자동화를 위해서는 작업물의 위치가 파악된다고 하더라도 작업을 할 수 있는 여러 기계로 인해 작업량을 할당해야 하며 또한 할당된 작업량을 각각의 gantry-robot이 처리할 수 있도록 최적화된 경로를 설정해 주어야 한다. 자동화 시스템에서는 이러한 절차를 자동으로 수행되도록 해야 한다. 따라서 다음의 3가지의 작업 계획 문제가 발생한다.

- § 3기의 gantry-robot에 작업량을 할당
- § 각각의 gantry-robot에 할당된 작업량을 용접하는 순서 결정
- § 각각의 seam에 대한 용접방향의 결정

이 모든 작업은 C++ 언어로 구현하였으며, dynamic linked library(.dll) 형태로 만들었다. 크게 다음과 같이 클래스를 생성하였다. (Fig. 3.1)

- ① AboutZone : 작업 영역에 관한 정보를 담고 있는 클래스
- ② City : Flat bar의 중심점 즉 node에 관한 정보를 담고 있는 클래스
- ③ Flat bar : 제품 내 Flat bar 즉, seam에 대한 정보를 담고 있는 클래스
- ④ Product : 제품에 대한 정보를 담고 있는 클래스
- ⑤ Chromo\_city : 유전자 알고리즘 연산 시 객 개체에 대한 정의 클래스
- ⑥ Pop\_city : 유전자 알고리즘 연산 시 모집단에 대한 정의 클래스

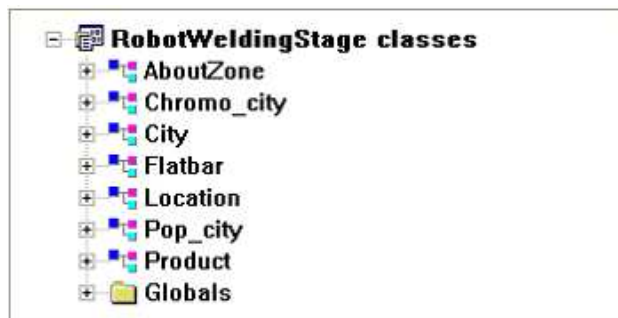


Fig. 3.1 Classes of Program



위에 제시된 작업 계획은 제품 정보로부터 시작한다. Fig. 3.2은 제품 정보가 담겨 있는 파일의 모습이다. 이 수치들은 각각 제품의 특징들을 표현하고 있다.

15						
2						
1	326990	245240	180			
3	340980	250160	90			
0	0					
19						
1	289600	289600	289600	288000	0	0
2	289600	288000	288800	287200	288800	288000
1	288800	287200	288800	245190	0	0
2	288800	245190	289800	244190	288800	244190
2	289800	244190	288800	243190	288800	244190
1	288800	243190	288800	-84480	0	0
1	288800	-84480	288800	-299910	0	0
2	288800	-299910	289300	-300410	288800	-300410
1	289300	-300410	289300	-302410	0	0
2	289300	-302410	288800	-302910	288800	-302410
1	288800	-302910	288800	-404650	0	0
1	288800	-404650	488800	-404650	0	0
1	488800	-404650	488800	-76980	0	0
1	488800	-76980	488800	243190	0	0
2	488800	243190	487800	244190	488800	244190
2	487800	244190	488800	245190	488800	244190
1	488800	245190	488800	287700	0	0
1	488800	287700	486900	289600	0	0
1	486900	289600	289600	289600	0	0
0						
2						
2A						
0						
430800	-403650	430800	288190			
2						
0	0					
-						
0						
359800	-403650	359800	288190			
2						
0	0					

Fig. 3.2 Product data

Fig. 3.3은 작업 계획의 절차를 나타내고 있는 순서도이다.

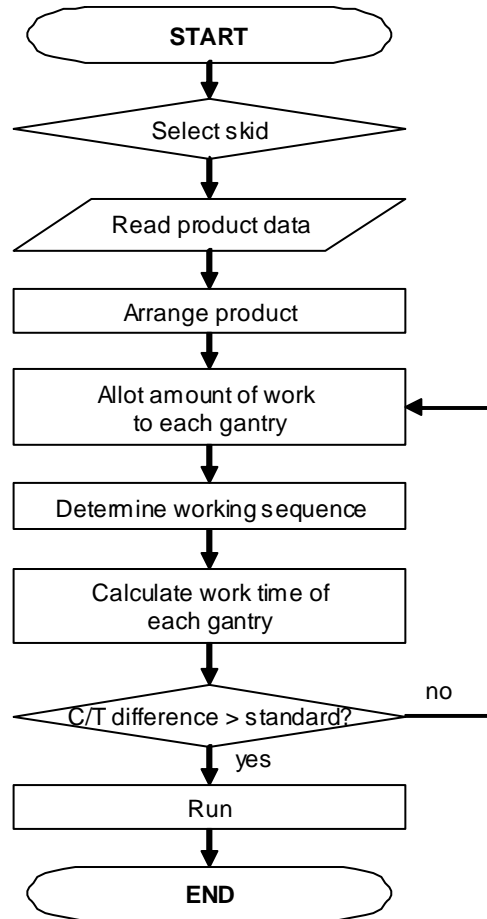


Fig. 3.3 Flow chart of planning

## 제 4 장 작업량의 배분

### 4.1 작업량 배분의 필요성

로봇용접장에는 총 6기의 gantry가 있으므로, 작업량은 6부분으로 나뉘야 한다.. Fig. 4.1은 로봇용접장에 놓인 부재들의 모습이다. 주판(Floor) 위에 소부재(Flat bar)가 용접되어 있음을 볼 수 있다. 로봇용접장에 들어오는 제품들은 그 형상이 서로 다르고, Flat bar의 위치 및 용접장이 다르다.



Fig. 4.1 Arrangement of product

따라서 Fig. 4.2와 같이 단순히 로봇용접장을 3등분하면 분배에 있어 불균형이 초래될 확률이 높다. 로봇용접장의 총 작업 시간은 여러 gantry-robot 중 작업 시간이 가장 긴 것의 작업 시간과 같게 된다. 물량의 고려 없이 작업 영역을 나누고 그것을 할당하게 되면 gantry-robot간 작업량의 불균형을 유발시켜 전체 작업 시간이 커지게 된다. 작업 시간은 3기의 gantry-robot의 작업 시간 중 가장 긴 시간에 의해 결정되기 때문이다. 따라서 3기의 gantry-robot이 동일한 작업을 갖도록 작업량을 배분해야 한다. Fig. 4.3의 3 geo는 단순히 로봇용접장을 라인 방향으로 3등분하고 그 물량을 3기의 gantry-robot에 할당한 결과 작업 시간을 보여주고 있다. Fig. 4.3의 even은 용접장을 기준으로 각각의 켄트리에 균등한 작업량을 배분한 경우의 작업 시간이다.

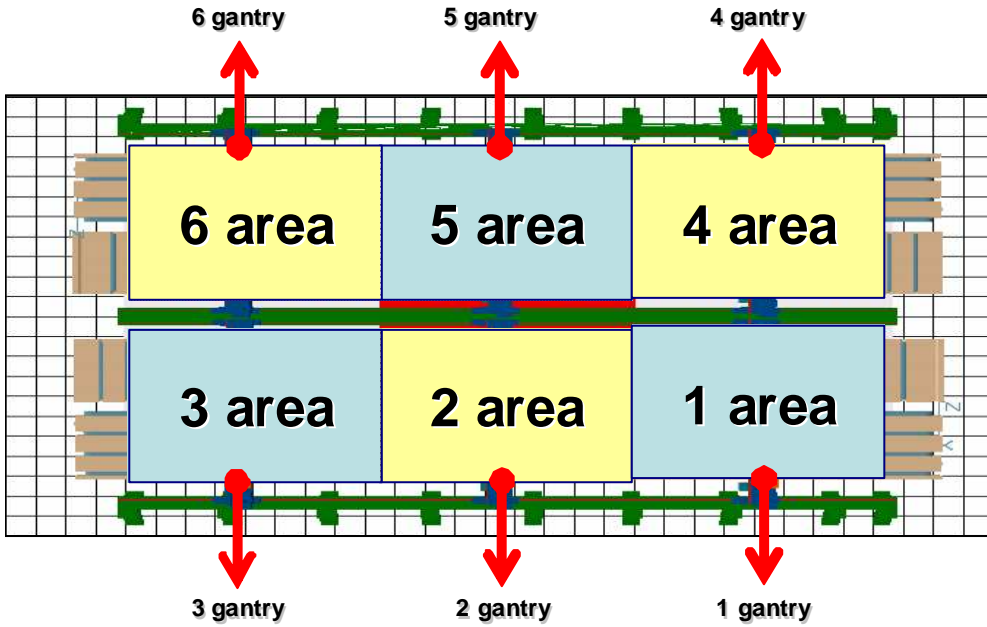


Fig. 4.2 Work area

Table 4.1 Cycle time for work area divided in 3 geometry

unit: sec

	3 geo	even		3 geo	even
1 area	33457.0	49014.3	4 area	37928.9	46814.3
2 area	54487.6	49138.8	5 area	45946.9	46829.1
3 area	59209.4	49000.9	6 area	54184.8	45047.2

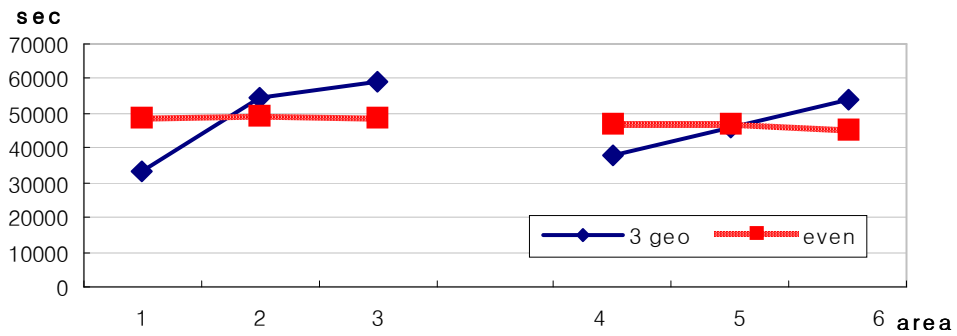


Fig. 4.3 Cycle time of 3 geo. and even distribution

Fig. 4.3에서 볼 수 있듯이 3 geo의 경우 작업 시간의 그래프가 구간 사이에 들쭉날쭉한데 반하여 even에서는 거의 균등한 것을 볼 수 있다.

## 4.2 작업량 배분 시 고려할 요소 및 목적 함수

작업 시간을 결정하는 것은 다음의 요소들로 나누어 생각할 수 있다.

### ① gantry-robot과 제품과의 거리

로봇용접장에 놓인 부재는 gantry-robot의 기수로 나뉘고 할당되는 작업량은 gantry의 초기 이동거리를 고려하여 할당되어야 한다. 각 gantry-robot은 서로 교차할 수 없다. Fig. 4.2에서 1area에 있는 제품은 1 gantry에 할당되어야 하지 3 gantry에 할당되어서는 안된다. 즉, 제품에 가까이 있는 gantry-robot에 작업량이 할당되어야 한다.

### ② 할당된 세 영역의 부재의 총 용접장

이 항목은 아래의 작업 시간 산정식에서 용접시간 및 티칭 시간과 관련된 부분이다. 작업 시간 중 가장 큰 비중을 차지하는 것은 용접시간이다. 따라서 용접장을 균등하게 하면 작업량의 차이를 줄일 수 있다.

### ③ 용접 경로

용접 경로는 seam사이를 이동하는 데 걸리는 시간 즉, gantry의 이동시간을 결정하는 요소이다. 용접 경로는 작업 시간 산정식에서 이동시간 항목에 해당하는 요소이다.

### ④ 용접 방향

용접 방향은 하나의 seam내에서 용접의 시작점과 끝점을 정함으로써 결정할 수 있으며, 이것은 이동 시간에 영향을 주는 요소이다.



Fig. 4.4 Welding direction

⑤ 로봇 모션 시간

gantry-robot이 작업을 수행함에 있어서 소요되는 부수적인 시간을 말한다. 로봇이 용접 및 이동을 위해 자세를 바꾸는 데 걸리는 시간, 와이어 커팅시간 등이 이에 해당되며 하나의 seam당 130초의 시간이 필요하며 이것은 작업 시간에 있어 큰 부분을 차지하고 있어 고려해야만 한다.

로봇 용접장의 공정상 작업 시간은 다음과 같이 표현된다.

$$\text{작업 시간} = \text{이동 시간} + \text{티칭 시간} + \text{이동 시간} + \text{용접 시간} + \text{로봇모션시간}$$

$$WT(\text{Working Time}) = 2 \times MD/MV + SL/TV + SL/WV + RMT$$

$$MD(\text{Moving Distance}) = \sum_i^{n-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}$$

$$WV(\text{Welding Velocity}) = 13.3m/s$$

$$MV(\text{Moving Velocity}) = TV(\text{Teaching Velocity}) = 150m/s$$

$$SL(\text{Seam Length}) = \sum_i^n \sqrt{(x_{s_i} - x_{e_i})^2 + (y_{s_i} - y_{e_i})^2}$$

$$RMT(\text{Robot Motion Time}) = 130s$$

### 4.3 작업량 배분 방법

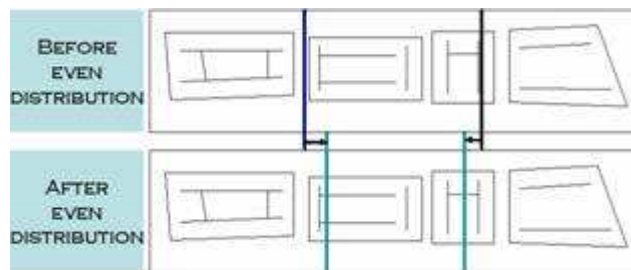


Fig. 4.5 Concept of distribution

Fig. 4.5는 작업량 배분 알고리즘의 개념도이다. 가상의 분할바를 이용하여 구간을 나누고 각 구간의 작업시간(working time zone) 및 구간 간의 작업 시간의 차

(working time difference)를 구한다.

$$\begin{aligned} \text{working time difference} = & \\ & | \text{working time\_eachzone}[1] - \text{working time\_eachzone}[2] | \\ & + | \text{working time\_eachzone}[2] - \text{working time\_eachzone}[3] | \\ & + | \text{working time\_eachzone}[3] - \text{working time\_eachzone}[1] | \end{aligned}$$

그 값이 기준치보다 작으면 분할이 끝나며 그렇지 않은 경우에는 분할바를 로봇 용접장 길이 방향을 따라 좌우로 분할바의 위치를 옮긴다. 다시 작업 시간 및 작업 시간의 차를 계산하며 기준치보다 작은 값을 가질 때까지 반복한다.

## 제 5 장 작업 순서의 결정

### 5.1 작업 순서의 중요성

하나의 gantry-robot은 많은 수의 flat bar를 용접해야 하며, flat bar를 어떠한 순서로 처리 할 것인지 결정해야 한다. Flat bar의 개수가 작으면 어떠한 순서로 작업을 하는 것이 시간을 줄일 수 있는 효율적인 경로인지 판단하기 쉽지만, 그 수가 많으면 경우의 수가 커지기 때문에 모든 경우를 고려하여 순서를 결정하는 일은 많은 시간과 계산 과정을 요구한다. 또한 자동화를 위해서는 flat bar개수의 많고 적음에 상관없이 작업 순서 설정을 위한 일정한 규칙이나 알고리즘이 필요하다.

작업 순서는 작업 시간과 직결되는 요소이므로 로봇의 동선을 짧게 하는 경로를 갖는 작업 순서의 경우는 작업 시간을 크게 줄일 수 있는 반면, 그렇지 못한 경로는 작업 시간을 늘리는 결과를 초래한다.

### 5.2 작업 순서 결정 문제의 특성

Gantry-robot에 할당된 flat bar의 용접 순서를 결정하는 일은 인공지능의 대표적인 문제인 TSP(Traveling Salesman Problem)로 간주 할 수 있다. TSP는 외판원 문제라고도 하며 외판원이 본점을 출발하여 n개의 도시를 단 한번씩 반드시 방문하고 다시 본점으로 돌아오는 최소거리(비용)의 경로를 찾는 문제이다. 외판원문제는 최소경로를 찾는 순열문제으로써  $n!$ 의 탐색공간을 갖는다.

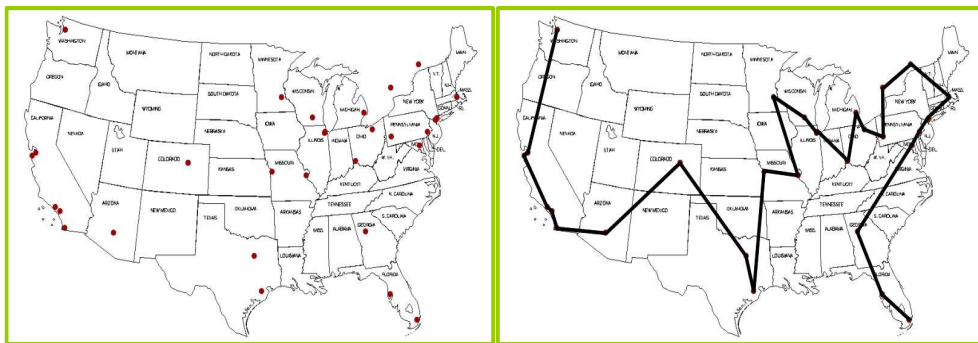


Fig. 5.1 Feature and solution of TSP



### 5.3 작업 순서 결정 방법의 모색

작업 순서를 결정하는 일은 TSP문제로 간주하였고, 이 문제를 어떤 방법으로 푸느냐는 문제가 남았다. Flat bar의 개수가 10개 정도만 되어도 작업 순서의 경우의 수는  $10! = 3628800$  이다. 경우의 수가 너무 많기 때문에 모든 경우를 고려하는 방법은 효율적이지 못하다. 즉 이것은 어떠한 최적화 기법을 사용할 것인가라는 문제와 같다.

기존의 대표적인 최적화 기법으로는 목적함수의 도함수를 이용하는 Gradient 기법이 있다. 그러나 이것은 설계변수가 많거나 목적함수가 복잡한 경우 최적해를 찾는 데 어려움이 있으며, 지역해(Local Optimum)에 빠질 가능성이 크다. 그러나 유전자 알고리즘은 기존의 최적화 방법과 달리 도함수의 정보가 필요하지 않으며, 지역해에 빠질 가능성이 낮고 해에 대한 신뢰성이 높기 때문에 많은 최적화 문제에 사용되고 있다.

### 5.4 유전자 알고리즘

#### 5.4.1 유전자 알고리즘의 개요

유전자 알고리즘은 생물의 진화과정, 즉 찰스 다윈(Charles Darwin)이 ‘種의 기원’에서 제시한 자연 선별(natural selection)과 멘델(Gregor Mendel)의 유전법칙을 모방한 확률적 탐색기법이다. 이 알고리즘은 1975년 Holland의 논문 "Adaptation in Natural and Artificial Systems"에서 처음 소개되었으며 자연시스템의 한 메커니즘으로 생물의 진화과정을 추상화하여 인공시스템을 설계하고자 했다. 그 후 20여 년 동안 유전자 알고리즘의 이론과 응용에 관하여 활발한 연구가 이루어져 왔다.

유전자 알고리즘의 가장 큰 특징은 뉴턴법과 같은 고전적 최적화기법이나 타부서치(tabu search), 시뮬레이티드 어닐링(simulated annealing)과 같은 이웃해 탐색기법이 하나의 해를 운용하는 데 반하여, 유전자 알고리즘은 복수 개의 잠재해들로 이루어진 해야 집단(population)을 운용한다는 것이다. 이러한 해집단에 자연선별과 유전법칙의 메커니즘을 적용하여 세대(generation)를 진행시키면서 해공간을 탐색해 간다. 유전자 알고리즘은 좋은 해의 이용능력과 해 공간의 탐색능력이 적절히 조화되어야 한다. 좋은 해의 이용만을 강조하면 조기 수렴하여 부분 최적에 빠질 수 있고 해 공간의 탐색만을 임의탐색에 가까워 좋은 해를 찾아가지 못하게 된다. 유전자 알고리즘은 매 세대마다 모집단을 운용함으로써 내재된 병렬성(parallelism)을 갖는 기법이다.

### 5.4.2 유전자 알고리즘의 요소

유전자 알고리즘은 문제의 잠재해를 표현한 개체들로 이루어진 모집단을 가지고 시작한다. 모집단은 매 세대마다 일정수의 개체를 유지하고 매 세대에서 각 개체의 적응도(fitness)를 평가하여 이에 따라 다음 세대에 생존할 개체들을 확률적으로 선별한다. 선별된 개체들 중 일부의 개체들이 임의로 짝을 지어 교배하여 자손을 생성한다. 이 때 교차(crossover)에 의해 부모의 유전자가 자손에게 상속되고 돌연변이가 일어날 수 있다. 자손은 부모로부터 좋은 유전형질을 상속받았다고 가정함으로써 다음 세대의 잠재해들은 평균적으로 전 세대보다 더 좋아진다고 본다. 이러한 진화과정은 종료조건을 만족할 때까지 반복한다.

유전자 알고리즘에서는 생물학의 용어가 자주 사용된다. 문제의 잠재해를 표현하는 개체는 스트링(string) 또는 염색체(chromosome)라 부르며, 각 개체를 이루는 단위를 유전자라 부른다. 어떤 형질을 갖는 유전자가 놓이는 위치를 염색체 좌(locos)라 하고, 각 유전자는 형질을 특징짓는 여러 상태를 가질 수 있는데 이를 대립유전자(allele)라 부른다. 유전자로 이루어진 염색체의 표현을 유전자형(genotype)이라 하고, 눈과 머리 색깔 등 보이는 생물의 체질(즉, 염색체가 의미하는 것)을 표현형(phenotype)이라 부른다. 유전자 알고리즘의 요소로는 표현, 초기 모집단, 적응도 평가, 선별, 유전연산자, 유전 파라미터 등이 있다.

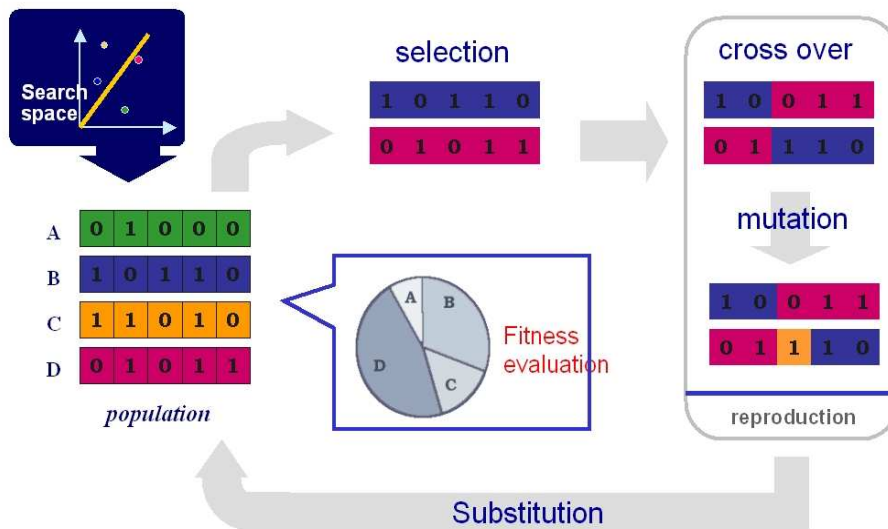


Fig. 5.2 General structure of genetic algorithm

### (1) 표현

유전자 알고리즘을 구축하는 첫 단계로 문제의 잠재해를 유전적 표현, 즉 개체로 표현하여야 한다. 이 유전적 표현은 유전자 알고리즘의 다른 절차(적응도 평가와 유전연산자 적용 등)에 영향을 주므로 문제의 특성을 잘 반영할 수 있어야 한다. 대표적으로 이진코딩법이 있으며 유전 연산자들은 이진 스트링을 기반으로 부호 공간상에서 연산되며, 개체 능력을 평가하는 목적함수에 적용 및 적합도 계산에서는 파라미터를 기반으로 해공간에서 이루어진다. 그 외에 순열 표현, 그룹번호 표현 등이 있다.

### (2) 초기모집단

유전자 알고리즘은 개체들로 구성된 모집단을 운영하므로 초기에 모집단이 생성되어야 한다. 초기의 모집단을 만드는 방법으로는 문제의 특성을 이용한 기존의 발견적 방법이나, 임의생성방법이 있다. 발견적 기법에 의한 초기해보다는 임의생성방법을 흔히 사용한다. 왜냐하면 발견적 기법에 의해 생성된 해들은 초기 수렴하여 해공간의 다양한 탐색을 방해하는 경향이 있기 때문이다.

### (3) 적응도 평가

적응도는 자연개체의 생존능력을 나타낸다. 최적화문제에서 적응도는 목적함수에 의해 측정된다. 적응도를 평가하는 함수로는 최적화문제의 목적함수 자체를 흔히 사용한다. 평가함수로 목적함수를 사용하는 경우, 개체  $v$ 의 평가함수값  $eval(v)=f(X)$ 가 된다. 여기서  $x$ 는 개체  $v$ 에 대응하는 변수 값이다. 그러나 평가함수로 목적함수를 사용하지 않고, 좋은 해를 유도할 수 있는 유전자 구조를 갖는 개체의 적응도가 높게 평가될 수 있는 함수를 사용하기도 한다.

### (4) 선별

선별은 적자생존의 자연법칙에 기초하여, 즉 환경에 대한 적응도에 의해 현 세대의 모집단으로부터 다음 세대에 생존할 개체를 선택하는 과정이다. 유전자 알고리즘에서 선별은 모집단의 다양성과 선별압력이 조화를 이룰 수 있어야 한다. 강한 선별압력은 모집단의 개체들을 초기 수렴시키는 경향을 갖는다. 즉 모집단의 다양성을 약화시켜, 해공간의 다양한 탐색을 막는 결과를 가져온다. 한편, 약한 선별압력은 모집단의 다양성은 유지되나, 좋은 해를 효율적으로 탐색하지 못하여, 임의탐색(random search)과 비슷한 결과를 초래할 수 있다. 선별방법은 확률바퀴(roulette wheel), 순위선별(ranking selection), 토너먼트 선별(tournament selection) 등 여러 방법들이 있다.

**(5) 유전 연산자**

유전연산자는 교차와 돌연변이로 나누어진다. 교차는 두 부모가 갖는 유전자를 조합하여 자손을 생산하는 과정이다. 교차는 좋은 해를 이용하는 역할을 한다. 이를 위해서 교차는 부모의 좋은 형질이 가능한 파괴되지 않고 자손에 상속될 수 있어야 한다.

■ Single point crossover



■ Two point crossover



Fig. 5.3 Single point crossover and two point crossover

돌연변이는 개체에 새로운 유전자가 생성되는 것으로, 한 개체에서 아주 작은 수의 유전자를 임의로 변화시키는 과정이다. 유전자 알고리즘에서 돌연변이는 해 공간을 다양하게 탐색하는 역할을 한다. 돌연변이는 완전히 새로운 유전자를 생성하기 때문에 이를 ‘진화의 연료’라 하기도 한다. 돌연변이는 대부분 해로운 것이다. 해로운 돌연변이는 자손들의 적응을 감퇴시켜 자연선택에서 제거된다. 그러나 때때로 유익한 돌연변이가 일어나며, 이런 돌연변이에 의한 개체는 환경에 잘 적응하여 유사한 자손을 번식시킨다.

■ 유전자를 일정한 확률로 변화시키는 조작



■ 개체군의 다양성 유지

→ local optimum 방지

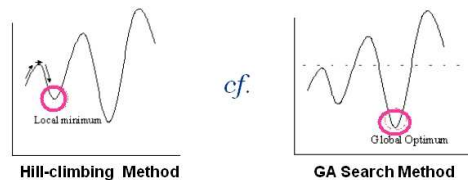


Fig. 5.4 Mutation

### (6) 유전 파라미터

유전자 알고리즘에 사용되는 파라미터로는 모집단의 크기, 교차율, 돌연변이율, 종료조건 등이 있다. 모집단의 크기는 모집단을 이루는 개체의 수를 의미한다. 교차율은 각 개체가 교차될 확률을 나타낸다. 모집단이 100개의 개체로 이루어진 경우에 교차율 0.5는 평균적으로 50개의 개체가 교차된다는 것을 의미한다. 돌연변이는 흔히 유전자 단위로 이루어진다. 돌연변이율은 각 유전자가 돌연변이 될 확률을 나타낸다. 예로 100개의 개체가 있고 각 개체가 10개의 유전자로 구성되었다면, 돌연변이율 0.01은 1,000개의 유전자 중에서 평균적으로 10개의 유전자가 돌연변이 한다는 것을 의미한다. 알고리즘의 종료조건으로는 진행된 세대수 또는 생성된 개체수, 해의 개선이 이루어지지 않고 진행된 세대수 또는 생성된 개체수, 계산 소요시간 등이 흔히 사용된다.

#### 5.4.3 유전자 알고리즘의 절차

```
begin
  t ← 0
  P(t)의 초기화 (초기모집단 생성)
  P(t)의 적응도 평가
  while (종료조건이 만족되지 않으면) do
    begin
      t ← t+1
      P(t-1)로부터 P(t)를 선별
      P(t)의 유전연산(교차와 돌연변이)
      P(t)의 적응도 평가
    end
  end
end
```

Fig. 5.5 Procedure of Genetic Algorithm

- $P(t)$  : 세대  $t$ 의 모집단
- $N_p$  : 모집단의 크기, 즉 모집단의 개체 수
- $N_g$  : 개체의 원소 수
- $P_c$  : 교차율
- $P_m$  : 돌연변이율

유전자 알고리즘은 다음과 같은 절차에 따라 진화한다.

단계 1: (초기모집단),  $t \leftarrow 0$ 으로 두고  $P(t)$ 를 생성한다. 즉, 초기모집단을 만든다.

단계 2: (적응도평가),  $P(t)$ 에 있는 모든 개체의 적응도를 평가한다.

단계 3: (선별),  $t \leftarrow t + 1$  로 두고  $P(t-1)$ 로부터  $P(t)$ 를 선별한다.

단계 4: (교차)

①  $[0,1]$ 사이의 난수를 발생시켜  $r < P_c$ 이면 그 개체를 교차하는 개체로 둔다. 모든 개체에 대해 이를 행한다.

② 교차 대상 개체 중에서 임의로 쌍(두 부모)을 만든다.

③ 각 쌍(두 부모)의 개체를 교차하여 두 자손을 생산한다.  $P(t)$ 에서 교차된 부모 개체를 제거하고 생산된 자손을  $P(t)$ 에 삽입한다.

단계 5: (돌연변이),  $P(t)$ 를 돌연변이시킨다. 개체의 각 원소에 대해  $[0,1]$ 사이의 난수  $r$ 을 발생하여,  $r < P_m$ 이면 그 원소를 돌연변이시킨다.

단계 6: (적응도 평가),  $P(t)$ 의 각 개체에 대해 적응도를 평가한다.

단계 7: (종료조건), 종료조건을 만족하면 끝낸다. 그렇지 않으면 단계 3으로 간다.

#### 5.4.4 스키마 정리

유전자 알고리즘의 이론은 스키마 정리(schema theorem)에 기초한다. 이 정리는 유전자 알고리즘에서 선별, 교차, 돌연변이 등의 일련의 과정에서 개체들에 존재하는 유사한 인자들이 어떻게 다른 개체로 전파 또는 소멸되면서 좋은 해를 찾아가는가를 보여준다. ‘스키마’는 개체들간의 유사성을 나타내는 일종의 전형(template)이다. 개체의 특정위치들에 존재하는 인자들의 값에 의해 스키마는 나타나며 스키마에서 특정인자의 값을 정의할 때 어느 인자값이나 가질 수 있다는 표시로 ‘\*’가 사용된다. 예로 이진표현에서 스키마(\* 0 0 1 1)에는 두 개체 (0 0 0 1 1)과 (1 0 0 1 1)이 속하고, 스키마 (\* \* 0 1 1)에는 네 개의 개체 (0 0 0 1 1), (0 1 0 1 1), (1 0 0 1 1), (1 1 0 1 1)이 포함된다. 따라서 이진표현에서  $r$ 개의 \*를 갖는 스키마에 부합되는 개체의 수는  $2^r$ 개가 된다.

##### (1) 차수와 정의 길이

스키마  $S$ 의 차수는 고정된 값을 갖는 인자(\*가 아닌 원소)의 수로 정의된다. 따라서 이진표현에서의 차수는 \*가 아닌 0또는 1의 값을 갖는 인자의 수가 된다. 차수는 스키마의 특이성을 나타낸다. 다음과 같이 개체의 길이가 10인 세 개의 스키마가 있다고 하자.

$$\begin{aligned}
S_1 &= ( * * * 1 0 0 * 1 0 0 ) \\
S_2 &= ( * * * * 0 1 * 1 * * ) \\
S_3 &= ( 0 1 1 0 * * * * * 1 )
\end{aligned}$$

스키마의 차수는  $\theta(S)$ 로 나타내며, 각 스키마의 차수는  $\theta(S_1) = 6$ ,  $\theta(S_2) = 3$ ,  $\theta(S_3) = 5$ 가 된다. 여기서 스키마 S1이 가장 큰 특이성을 갖는다고 할 수 있으며, 스키마 S1에 일치하는 개체의 수가 스키마 S2나 S3에 비해 적다는 것을 의미한다. 스키마의 차수 개념은 어떤 스키마가 돌연변이에 의해 생존할 확률을 계산하는데 사용된다.

스키마의 정의길이는 개체의 고정된 위치들의 처음과 끝 사이의 거리로 정의되며, 스키마 S의 정의길이를  $\delta(S)$ 로 나타내자. 이것은 스키마에 포함된 정보의 밀집성을 정의한다. 위의 각 스키마의 정의 길이는  $\delta(S_1) = 10-4 = 6$ ,  $\delta(S_2) = 8-5 = 3$ ,  $\delta(S_3) = 10-1 = 9$ 가 된다. 정의길이의 개념은 어떤 스키마가 교차에 의해 생존할 확률을 계산하는데 사용된다.

## (2) 선별의 효과

스키마의 한 특성은 t세대에서 스키마 S의 적응도  $eval(S, t)$ 로 나타낼 수 있다.  $eval(S, t)$ 는 스키마 S를 갖는 모집단 개체들에 대한 적응도의 평균으로 정의된다. t세대에서 스키마 S에 속하는 개체가  $\{v_{i1}, v_{i2}, \dots, v_{ip}\}$ 로 p개 존재한다면

$$eval(S, t) = \sum_{j=1}^p eval(v_{ij})/p$$

가 된다. 선별단계에서는 적응도에 의해 모집단의 수만큼 전 세대에 생성한 모집단으로부터 개체를 선택한다. 선별 후  $m(S, t+1)$ 은

$$\text{스키마 S에 속하는 개체의 평균 선별확률} : eval(S, t)/F(t)$$

( $F(t)$ ): t세대의 전체 모집단에 대한 적응도의 총합)

$$\text{이전 세대에서 스키마 S에 속했던 개체수} : m(S, t)$$

$$\text{모집단의 크기} : N_p \text{ or } n$$

등의 세 요소에 의해 결정된다. 즉

$$m(S, t+1) = m(S, t) \cdot n \cdot eval(S, t)/F(t)$$

로 표현된다. 그리고 모집단의 평균적응도  $F(t) = F(t)/n$  이므로 선별 후에 모집단에서 스키마 S를 갖는 개체의 수는 다음과 같이 된다.

$$m(S, t+1) = m(S, t) \cdot eval(S, t)/\bar{F}(t)$$

이 식을 재생산 스키마 성장식(reproductive schema growth equation)이라 부른

다. 이 식으로부터  $m(S, t+1)$ 는 모집단의 평균적응도에 대한 스키마의 적응도 비율로 결정됨을 알 수 있다. 이것은 평균 이상의 적응도를 갖는 스키마는 다음 세대에서 개체수가 증가하고, 평균 이하의 적응도를 갖는 스키마는 그 수가 감소하며, 적응도가 평균인 스키마는 그 상태를 유지한다는 것을 의미한다.

스키마 S가  $\epsilon\%$ 만큼 평균 이상의 적응도를 갖는 경우  $m(S, t) = m(S, 0) \cdot (1 + \epsilon)^t$  이고,  $\epsilon = (eval(S, t) - \bar{F}(t))/F(t)$  이 된다. 따라서 적응도가 평균 이상인 스키마를 갖는 개체의 수는 다음 세대에서 지수적으로 증가하게 된다.

### (3) 교차의 효과

스키마의 정의 길이가 그 스키마의 생존과 파괴확률에 중대한 역할을 하고 있다. 일점교차연산에서 절단점은 일반적으로  $l-1$ 개의 가능한 위치 중에서 임의로 선택된다. 이것은 일점교차에 의한 스키마 S의 파괴확률이  $P_s(S) = 1 - \frac{\delta(S)}{l-1}$  이라고 할 수 있다. 또한 모집단의 모든 개체들이 교차되는 것이 아니라 교차에 선택된 개체들만이 연산되고, 교차율이  $P_c$ 이므로 교차에 의한 스키마 S의 생존확률은  $P_s = 1 - P_c \cdot \frac{\delta(S)}{l-1}$  이 된다. 그런데 교차점이 스키마의 고정위치 사이에 선택되더라도 아주 적기는 하지만 그 스키마가 생존할 가능성이 있다. 따라서 교차에 의한 스키마 S의 생존확률은 다음과 같이 변형된다.

$$P_s(S) \geq 1 - P_c \cdot \frac{\delta(S)}{l-1}$$

결국 선별과 교차에 의한 재생산 스키마성장식은 다음과 같이 된다.

$$m(S, t+1) \geq m(S, t) \cdot eval(S, t) / \bar{F}(t) \cdot \left[ 1 - P_c \cdot \frac{\delta(S)}{l-1} \right]$$

이 식은 다음 세대에서 스키마 S에 부합되는 개체수가 이전세대에 그 스키마에 속했던 개체수, 스키마의 상대적 적응도, 그리고 정의길이의 함수임을 의미한다. 이 식으로 짧은 정의길이와 평균 이상의 적응도를 갖는 스키마가 선택될 확률은 세대가 진행됨에 따라 지수적으로 증가한다는 것을 알 수 있다.

### (4) 돌연변이의 효과

유전자 알고리즘에서 돌연변이는  $P_m$ 의 확률로 각 개체 내 각각의 원소들을 선택하여 임의로 변경한다. 이진 표현의 돌연변이에서는 흔히 0을 갖는 원소는 1로, 1을 가질 때는 0으로 변경한다. 한 스키마가 돌연변이에 생존하기 위해서는 스키



마의 모든 고정위치들이 변화되어서는 안된다. 다시 말해, 고정위치들 중 하나라도 돌연변이되면 그 스키마는 파괴된다. 따라서 돌연변이의 영향은 그 스키마의 차수에 따라 달라진다. 또한 한 원소가 변경될 확률은  $P_m$ 이므로 각 원소가 돌연변이 후에 생존할 확률은  $1 - P_m$ 이 된다. 그리고 하나의 돌연변이는 다른 돌연변이와 독립적이므로 스키마 S가 돌연변이에 의해 생존할 확률  $P_s(S) = (1 - P_m)^{o(S)}$ 가 된다. 흔히 돌연변이율은  $P_m \ll 1$ 의 아주 작은 값이 사용되므로

$$P_s(S) \approx 1 - o(S) \cdot P_m$$

결국 스키마 S에 대한 선별, 교차, 돌연변이의 결합된 효과는 아래와 같은 재생산 스키마성장식으로 나타낼 수 있다.

$$m(S, t+1) \geq m(S, t) \cdot eval(S, t) / \bar{F}(t) \cdot \left[ 1 - P_c \cdot \frac{\delta(S)}{l-1} - o(S) \cdot P_m \right]$$

이 식은 선별, 교차, 돌연변이를 거쳐 다음 세대에 스키마 S에 부합되는 개체수가 이전단계의 스키마수, 스키마의 상대적 적응도 그리고 정의길이와 차수의 함수임을 나타내고 있다. 또한, 짧은 정의 길이와 낮은 차수를 갖는 적응도가 평균 이상인 스키마가 다음 세대에 선택될 확률은 지수적으로 증가한다는 것을 의미한다.

이러한 스키마 이론에 근거한 유전자 알고리즘은 세대가 갈수록 좋은 해들을 유도한다고 설명할 수 있다.

## 5.5 유전자 알고리즘의 적용

본 연구에서는 각 Flat bar의 중심점을 node로 보고 겐트리가 이 node를 거쳐 돌아오는 최단 경로를 찾는 문제로 변환하였고, 여기에 유전자 알고리즘 기법을 적용하였다. Fig. 5.6 은 그 개념을 보여주고 있다.

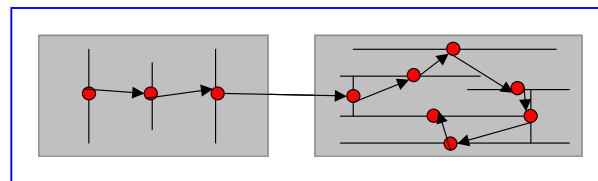


Fig. 5.6 The concept of application

### 5.5.1 표현형의 설정

작업의 순서를 위한 해를 얻기 위해 어떠한 개체를 운용하는 것이 가장 효율적

인지를 고려하여 표현형을 설정하였다. 본 문제의 표현형을 이진 코드로 할 경우 경로 표현에 있어 어려움이 있으며 계산과정 또한 복잡해진다. 따라서 각각의 flat bar에 알파벳으로 이름을 붙이고 그것의 순서를 작업 순서로 표현 할 수 있도록 하였다. 표현형은 Fig. 5.7과 같은 형태로 나타난다. 표현형에서 각 인자가 나타나는 순서가 gantry-robot의 작업의 순서가 된다.

( A B C D E F G H I J K L M N O ) ( B G H C M N K J L O E F A I D )
--

Fig. 5.7 The string of solution

### 5.5.2 초기모집단

초기모집단은 flat bar의 개수만큼 난수를 발생시켜 다양한 초기해를 설정하도록 하였다. 개체는 순서를 표현해야 하므로 개체 내 인자는 중복 없이 한번씩 모두 나타나도록 설정해야 한다.

### 5.5.3 적응도 평가

적응도 평가를 위해서는 각각의 개체가 목적하는 바에 얼마나 부합하는가를 판단해야 하기 위한 기준이 있어야 한다. 이것을 목적함수라고 하며 목적함수는 모든 node를 경유하는 거리를 최소로 하는 함수이다. Flat bar의 중심점의 좌표값을 이용하여 각 개체 및 모집단에 대한 적응도를 구하여 다음 세대로의 개체 선별에서 사용할 수 있도록 하였다.

### 5.5.4 선별

선별과정에서는 목적함수를 통해 얻어진 목적함수값을 이용하여 개체를 다음 세대로 가져오도록 하였다. 여러 가지의 선별 방법 중에서 확률바퀴방법을 사용하였다. 확률바퀴방법은 가장 대표적인 선별방법으로 개체의 적응도에 비례하여 개체가 선택될 확률을 부과하는 것이다. 따라서 각 개체에 대하여 목적함수를 통해 작업 시간을 구하고 목적함수 값이 작은 개체, 즉 이동거리가 짧은 개체가 선택되도록 하였다.

Table 5.1 Selection probability by roulette wheel

번호	개체	$x$ 값	적응도 : $f(x)$	선별확률
1	( A B D C E F )	10	2	0.047
2	( D C B A F E )	2	10	0.233
3	( D A F E C B )	30	22	0.511
4	( B C A E F D )	17	9	0.209
합계			43	1.000

### 5.5.5 교차

선별이 된 개체들로 새로운 집단이 구성되면, 교차단계를 거친다. TSP문제에서는 어느 도시에서 어느 도시로 가는지에 관한 즉, 인접정보가 중요하므로 인접인자재결합교차(edge recombination crossover)를 사용하였다. 이 교차는 인접정보를 중요시한다. 외관원문제에서는 어느 도시에서 어느 도시로 가느냐가 중요한 정보가 될 것이다. 따라서 외관원문제와 같이 인접한 인자의 정보가 중요시되는 경우에는 인접인자재결합교차의 사용이 바람직하다.

Table 5.2 Adjacent gene

flat bar	adjacent flat bar
A	I, B, D
B	A, C, E
C	B, D, F
D	C, E, H, A
E	D, E, I, B
F	E, G, C
G	F, H
H	G, I, D
I	H, A, E

단계 1: 시작인자(flat bar)를 선택한다. 한 부모의 첫인자를 선택하거나 인접표에서 인접한 flat bar의 수가 가장 적은 도시를 선택할 수 있다. 선택된 flat bar는 인접표에서 지운다.

단계 2: 직전에 선택된 flat bar와 연결된 flat bar 중에서 남아있는 인접된 flat bar의 수가 가장 적은 flat bar를 선택한다. 동일한 경우 임의로 선택한

다.

단계 3: 완전히 새로운 자손이 생성될 때까지 단계 2를 반복한다.

### 5.5.6 돌연변이

돌연변이 연산자로는 교환연산자를 사용하였다. 교환연산자는 부모에서 서로 같이 않은 임의의 두 인자를 교환하여 자손을 생산하는 방법이다.

돌연변이 전 : ( A B C D E F G H I J K L M N O )

돌연변이 후 : ( A B C D L F G H I J K E M N O )

### 5.5.7 종료 조건

종료 조건으로는 진행된 세대수를 사용하였다. 보통 하나의 gantry-robot에 할당되는 flat bar는 15~30개이며 표현형의 인자가 많기 때문에 세대수가 짧게 되면 충분한 연산을 수행하지 못하여 좋은 해를 유도하지 못한다. 그러한 문제로 인해 1500세대까지 해를 진화시켰다. 유전자 알고리즘에서는 선별과정이 확률적으로 진행되기 때문에 좋은 해가 다음 세대에 유전되기 위해 선택되지 않을 수 있고, 또한 좋은 해가 유전연산 과정에서 파괴될 수도 있다. 이런 현상을 방지하기 위해 각 세대가 지나는 동안 그 때까지 나타난 가장 좋은 해를 따로 보관하는 방법 (elitist strategy)을 사용하였다. 이렇게 하면 가장 좋은 해가 선별과정에서 사라지지 않고 계속 유지될 수 있다. Fig. 5.8은 세대가 진행됨에 따라 해가 진화해가는 것을 보여준다.

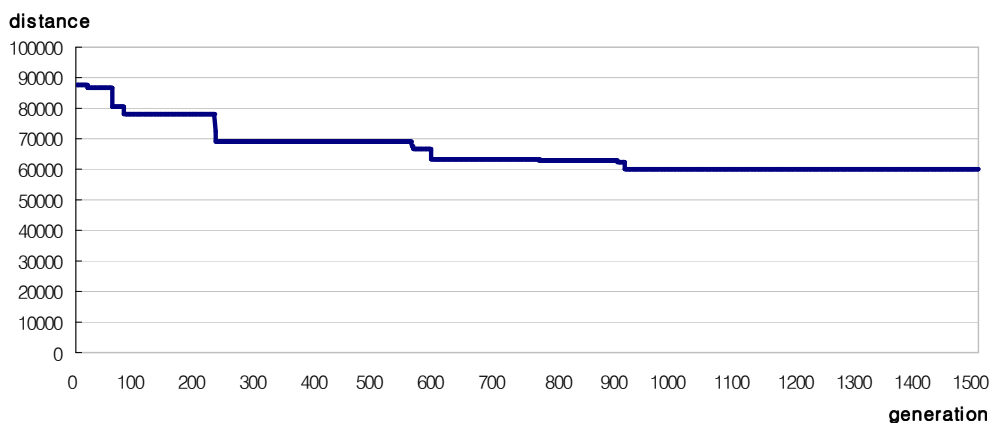


Fig. 5.8 Improvement of solution by generation

## 제 6 장 용접방향의 결정

작업 순서가 결정되면 gantry-robot이 flat bar를 거치는 순서가 결정된다. 그러나 flat bar의 중심을 node로 삼았기 때문에 flat bar의 양 끝단 중 어느 점이 용접의 시작점인지는 알 수 없다. 즉, 용접 방향을 결정해야 한다. 용접 방향을 결정하는 방법은 Fig. 6.1과 같다.

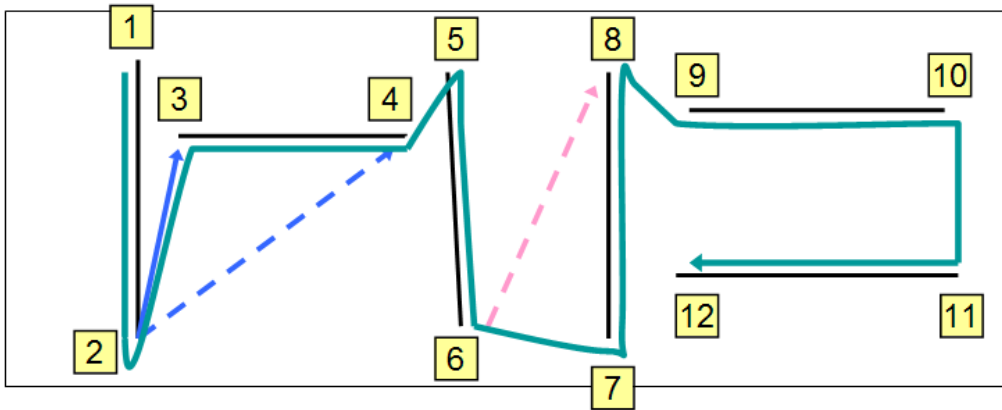


Fig 6.1 Concept of algorithm for welding direction

Flat bar[1-2]에서 flat bar[3-4]로 이동할 경우 flat bar[1-2]의 용접끝점에서 flat bar[3-4]의 두 끝단과의 거리를 각각 구한다. 위치 2에서 Flat bar[3-4]로 가는 두 가지 길이 있으며 그것은 실선과 점선으로 표시했다. 그리고 그 거리가 짧은 쪽(실선)을 flat bar[3-4]의 용접 시작점으로 하고 다른 끝단은 용접 종료점으로 한다. 이러한 방식으로 용접방향이 결정된다. 이 작업을 마지막 flat bar까지 반복함으로써 모든 flat bar에 대하여 용접 방향을 결정할 수 있다. 실제 현장에서 작업 시에는 이론적으로 알고 있는 용접 변형을 고려한 순서 및 방향을 따르지 않는다. 그것보다 작업자의 경험을 위주로 가능한 빨리 작업을 할 수 있는 순서 및 방향을 따른다.

## 제 7 장 작업계획을 고려한 시뮬레이션

### 7.1 시뮬레이션이란 무엇인가

시뮬레이션은 공정에 있어서 실제 시스템을 관찰하기 위한 최고의 도구이다. 시뮬레이션을 통해 수집된 데이터는 시스템을 설계하는데 사용된다. 시뮬레이션은 최적화 기법이 아니며, 그것은 모델링된 시스템의 수행의 정도를 판단 및 추정하기 위한 기법이다.

3차원 모델은 실세계를 3차원의 가상공간으로 특징화하여 옮긴 것을 말한다. 3차원 공간 속의 로봇, 겐트리, 부재 또는 공장은 실제와 그 형상이 같다. 그리고 모델은 한계속도, 한계각, 가용범위 등의 실제 모델의 속성을 그대로 갖고 있으며, 그것의 기능 및 모션 또한 실제와 같다. 따라서 검증단계(Validation)를 거친 3차원 모델은 실제 모델로 간주되며 이를 대상으로 구체적인 사안에 대한 시뮬레이션을 수행하게 된다. 소조립 공정에 있어 3차원 시뮬레이션의 주요 장점은 무엇보다도 생산 공정을 그대로 보여주는 가시화에 있다.

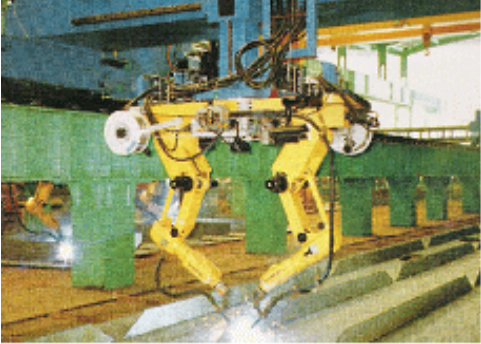
### 7.2 3차원 시뮬레이션의 선택

여러 설비와 공정이 상호적으로 작용하는 시스템에서 기존의 공정이나 설비를 변경하고자 하는 경우 예상되는 결과를 확인해보아야 한다. 여기에는 가능성을 가진 여러 방법이 있으며 그 중 3차원 공간에서의 모의 시험하는 방법을 생각할 수 있다. 3차원 공간에서의 시뮬레이션은 대상을 가상환경으로 그대로 옮겨 놓은 것이므로 실제와 동일하다고 간주할 수 있으며, 수치적인 방법으로는 파악하기 힘든 설비 간 충돌, 간섭 또한 쉽게 찾아 낼 수 있다. 특히, 하나의 제품이 완성되기까지 많은 공정을 필요로 하는 경우, 시·공간상으로 광범위한 경우에는 공정사이의 상관관계를 파악하기가 힘들며, 전단계 공정에서의 변화가 후공정에 미칠 영향을 예측하기 힘들다. 만일 용접속도가 향상되었다 가정하고 전후 공정간, 각각의 공정에 어떠한 영향을 미칠지를 알아보자. 용접속도가 빨라졌기 때문에 전체적인 작업 시간도 단축되리라는 것은 예상할 수 있지만 그러한 정성적인 결과는 시스템 개선 시 실제로 적용하기에는 부족하다. 3차원 공간에 탐색대상을 실제와 똑같이 구현하고 시뮬레이션을 하게 되면 정량적인 결과값을 얻을 수 있으며 이를 공정 계획에 반영할 수가 있다.

### 7.3 로봇 용접 공정의 3차원 시뮬레이션

3차원 시뮬레이터를 통해 작업 계획의 타당성을 검증하고 스키드에 대한 작업 시간을 추정할 수 있다. PPR의 개념에 따르면 **Product**는 제품, **Process**는 공정, **Resource**는 로봇용접장(Shop), 설비이다. 공장을 3차원으로 옮기기 위해서는 이 모든 것이 3차원으로 구현되어야 한다. 먼저 Product는 고정된 것이 아니라 매회 변하므로 모든 경우에 적용할 수 있는 함수를 구성하여 virtual shop내에 생성하도록 하였다. Process는 외적으로 완성된 작업장에 실제 겐트리나 로봇의 구동방식을 추가함으로써 구현된다. Resource는 CATIA를 이용하여 모델링을 하였으며, Resource에 대한 도면 및 사진을 이용하여 형상을 모델링하고 이들을 이용하여 IGRIP이라는 툴을 이용하여 Layout을 생성하였다. 이렇게 만들어진 공장은 'virtual shop'이라하고 실제 공장을 'real shop'이라 칭한다(Fig. 7.3). Table. 7.1은 로봇 용접 공정의 용접 로봇과 사양이며, Fig. 7.2는 로봇 프로세스가 virtual shop에 어떻게 적용되는지를 보여주는 그림이다.

Table 7.1 Robot specification

	<b>AS-1 Robot Specification</b>		
	Type	<b>Vertical articulated</b>	
	Weight	<b>50 kg</b>	
	Max. Payload	<b>6 kg</b>	
	Axis	<b>6 d.o.f</b>	
	Servo Drive	<b>Electric servo drive by AC servo encoder</b>	
	Position Detection	<b>Incremental encoder</b>	
<b>Travel &amp; Maximum Speed</b>	Axis	Travel	Maximum Speed
	1 Axis	<b>± 150 deg</b>	<b>112.5 deg/s</b>
	2 Axis	<b>± 90 deg</b>	<b>112.5 deg/s</b>
	3 Axis	<b>+160 ~ -150 deg</b>	<b>112.5 deg/s</b>
	4 Axis	<b>± 140 deg</b>	<b>180 deg/s</b>
	5 Axis	<b>± 120 deg</b>	<b>150 deg/s</b>
	6 Axis	<b>± 180 deg</b>	<b>180 deg/s</b>

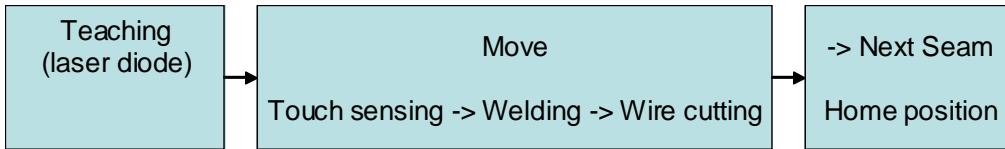


Fig. 7.1 Welding process

Table 7.2 Robot position

ROBOT POSITION	SPEED
ready0 # (0,-178,246,0,67,90)	
ready1 # (0,-175,225,0,-20,0)	0->1 speed 200mmps
ready2 # (0,-175,242,0,-42,0)	1->2 speed 400mmps
ready3 # (0,-122,188,0,-42,0)	2->3 speed 500mmps
0 -> 1 -> 2 -> 3 -> WELDING -> 3 -> 2 -> 1	

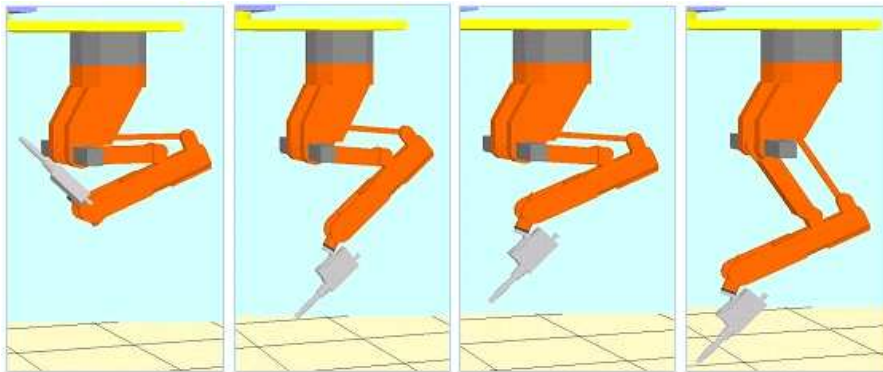


Fig. 7.2 Application of robot position

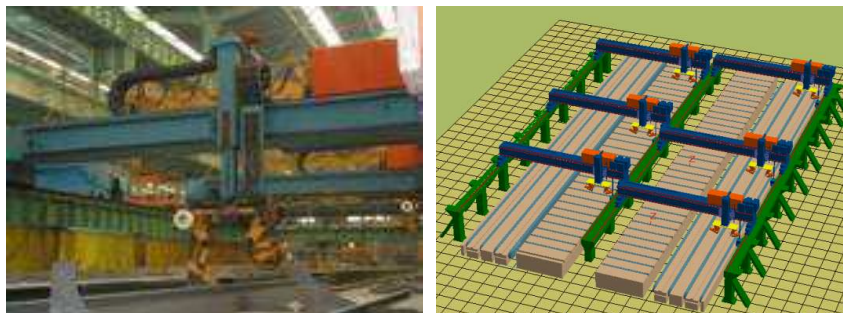


Fig. 7.3 Real shop and virtual shop



시뮬레이션부분은 크게 두 부분으로 나눌 수 있다. 먼저 machine(젠티리, 로봇)이 움직이기 전 부재생성, 배치, 작업 계획에 해당하는 로봇 모션이 들어가기 전 준비 단계이다. 이것은 물량이 바뀔 때마다 달라진다. 다른 하나는 실제로 machine이 움직임을 갖고 작업을 수행하는 부분이다. 따라서 우리는 첫 번째 부분은 macro를 만들어 자동으로 실행되도록 하였으며, 두 번째 부분은 macro를 통해 생성된 데이터를 활용하여 로봇 모션을 위한 함수의 매개 변수로 사용하도록 하였다.

지금부터 macro 내의 기능에 대해 macro실행순서대로 자세히 설명하겠다.

7.3.1 로봇 용접 공정 시뮬레이터(Planning)

(1) 시뮬레이터는 제품 정보를 이용하여 virtual shop에 workpiece를 생성한다.

Fig. 7.4는 부재의 제품 정보와 이를 통해 생성된 부재의 형상을 보여준다. 이 제품 정보의 각각은 모두 제품의 형상 및 작업 조건을 나타낸다. 이 데이터들은 floor내의 flat bar의 용접 위치, 두께, 길이 등의 형상정보로 사용되었다.

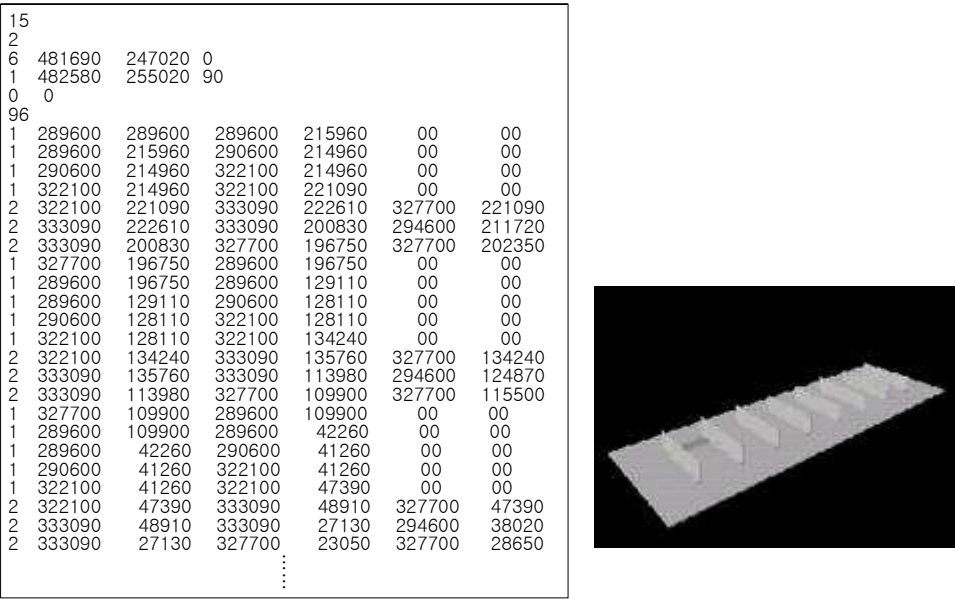


Fig. 7.4 Floor and flat bar

(2) 시뮬레이터는 로봇용접장의 컨베이어 위에 제품을 배치시킨다.

현재는 제품 배치를 위한 스킵 패턴이 없으며 배치는 작업자에 의해 이루어진

다. 작업자는 크레인을 이용하여 작업장 내에 제품을 배치시키며, 이것은 작업자의 경험에 따른 의사결정에 의해 이루어진다. 즉 제품의 배치에는 특정한 규칙이 없다. 그러나 virtual shop내에서는 배치를 위한 규칙이 필요하며, 규칙은 다음과 같이 정의한다.

- 규칙 1 : 제품은 일정한 간격(0.5m)으로 컨베이어 위에 놓인다.
- 규칙 2 : 제품은 로봇용접장 길이 방향을 따라 놓인다.
- 규칙 3 : 제품이 한 라인에 가득 놓이면 다음 라인으로 옮긴다. (Fig. 7.5)

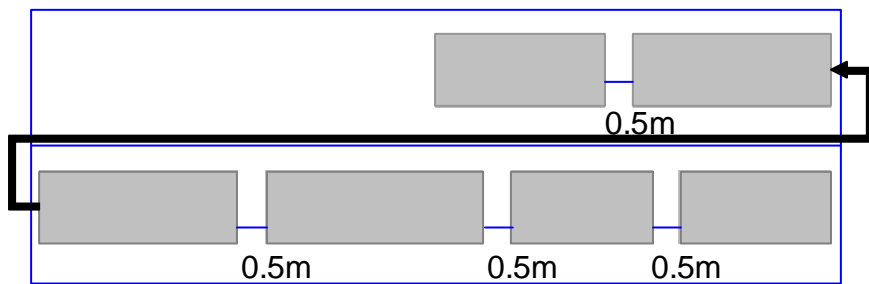


Fig. 7.5 Arrangement of products

**(3) 시뮬레이터는 작업 영역을 결정하고 겐트리에 할당한다.**

작업장의 겐트리의 수가 6이므로, 작업 영역은 6구역으로 나누어 각 겐트리에 할당하여야 한다. 시뮬레이터는 제품이 놓인 위치와 용접장을 계산하여, 각각의 겐트리에 작업량을 배분한다. Fig. 7.6은 작업 영역 할당의 한 예이며, 이것은 공간을 3등분하여 작업 영역을 설정한 것이다.

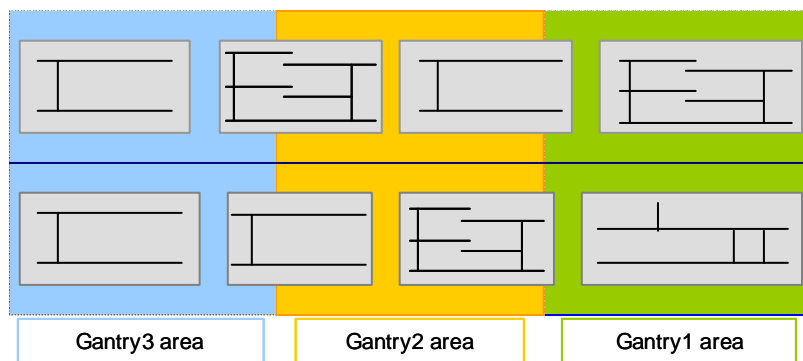


Fig. 7.6 Determination of the work area

**(4) 시뮬레이터는 작업 순서를 결정한다.**

실제로, 작업 순서는 teaching pendant를 갖고 있는 작업자에 의해 결정되며 알고리즘화 된 규칙이 없다. 그러나 virtual shop에서는 작업 순서를 위한 규칙이 필요하다. 작업 순서는 제품이 놓인 위치를 중심으로 결정된다. 작업 영역 설정 시 작업 순서까지 고려하도록 하였다. 결정된 작업 순서는 겐트리에게 알려준다.

**(5) 시뮬레이터는 용접 방향을 결정한다.**

Flat bar의 용접 시작점의 위치 값이 종료점의 위치 값보다 작은 경우는 0으로, 그렇지 않은 경우는 1로 하여 표시하고, 작업 영역 설정 시 이를 고려하도록 하였다.

**(6) 시뮬레이터는 (1)~(5)의 결과를 이용하여 겐트리 및 로봇의 움직임을 위한 인자를 생성한다.**

겐트리에 할당된 작업량, 작업 순서 및 방향들은 모두 시뮬레이션을 위해 필요한 정보들이며 이들을 데이터화 하여 시뮬레이터 내의 machine의 모션 구현 시 사용할 수 있도록 한다.

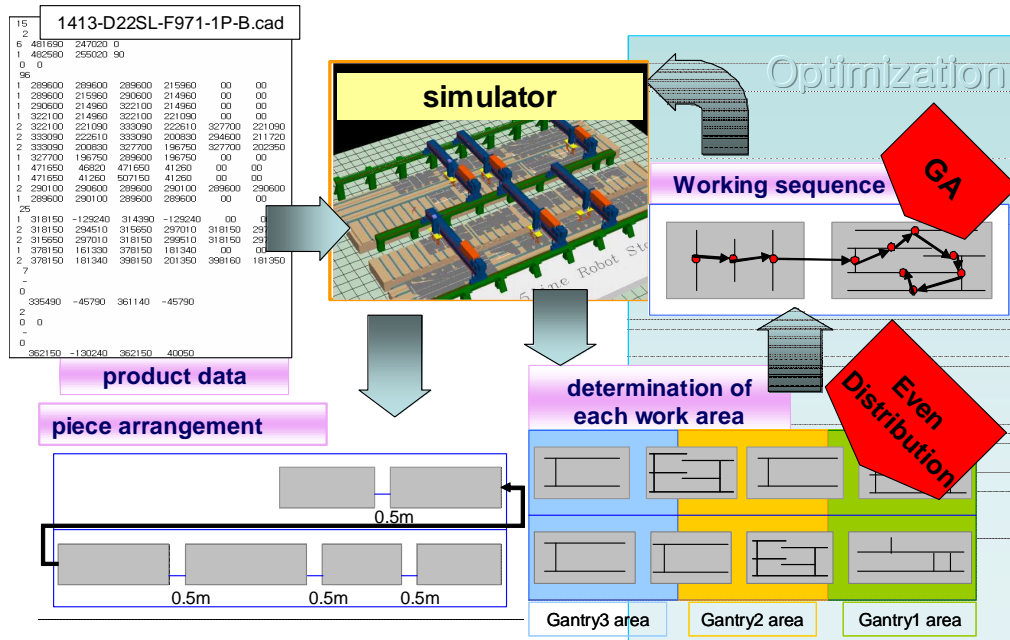


Fig. 7.7 Procedure of simulation

### 7.3.2 로봇 용접 공정 시뮬레이터(Motioning)

Fig. 7.8은 7.3.1의 결과로 생성된 결과 파일의 모습이며, 이러한 형태의 파일이 겐트리의 수만큼 생성된다. 각 겐트리 및 로봇 프로그램은 이 파일을 읽어 겐트리 및 로봇을 구동시킨다. 이러한 데이터에 따른 움직임이 끝나면 시뮬레이터는 작업 시간 즉, cycle time을 산출한다.

```
10
1420-B12DM-G23-1P-P.cad
1
1420-B12DM-G23-1P-P.cad
1
1420-B12DM-G23-1P-P.cad
1
1420-B12DM-G23-1P-P.cad
1
1420-B12DM-G23-1P-P.cad
1
1420-B12DM-F21-1P-P.cad#7
1
1420-B12DM-F21-1P-P.cad#7
1
1420-B12DM-F21-1P-P.cad#6
1
1420-B12DM-F21-1P-P.cad#6
1
1420-B12DM-F21-1P-P.cad#6
1
pth_1_1
pth_1_2
pth_1_3
pth_1_4
pth_1_5
pth_1_6
pth_1_7
pth_1_8
pth_1_9
pth_1_10
```

Fig. 7.8 Output file of planning

#### 7.4 작업 계획의 적용

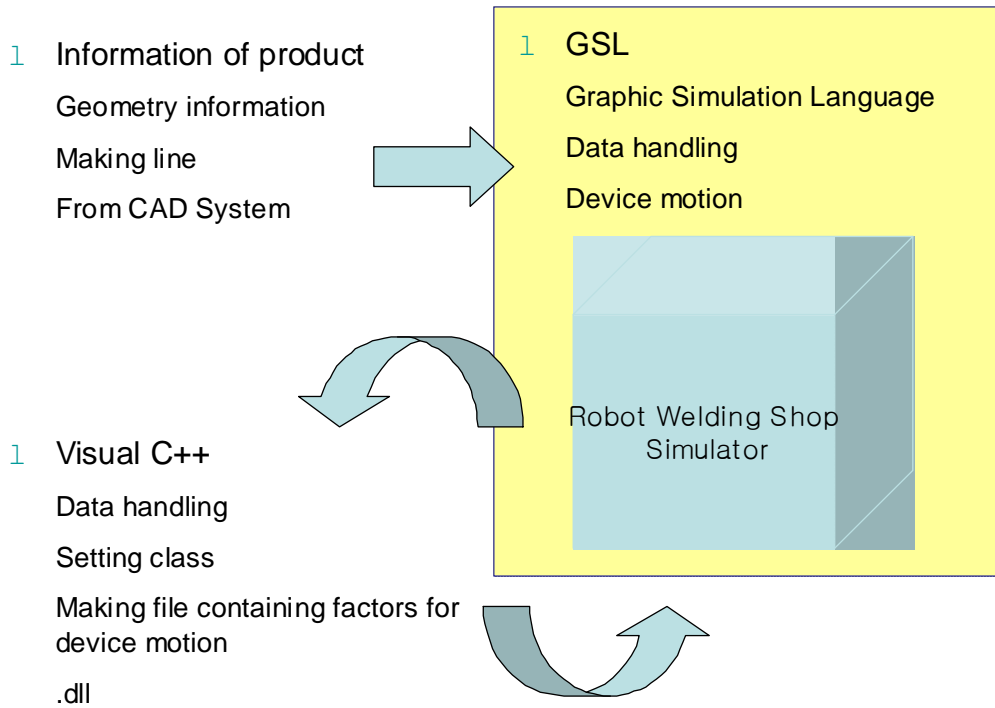


Fig. 7.9 Application of planning

#### 7.5 시뮬레이션 조건

하루의 물량을 대상으로 시뮬레이션하였다. 로봇용접장에는 하루 4회 물량이 들어오며 1회 물량은 1개 이상의 스킵으로 구성되어 있다. Fig. 7.10은 소조립 공정의 스케줄 표를 보여주며 특정일에 라인에 투입되는 스킵 및 시간, 용접장 및 시수에 대한 정보를 얻을 수 있다.

BAY	호선	블록	SKID	용접장			작업별 시수					일정정보					
				공작도	구조도	M	자동 요건	배재	취부	용접1	용접2	1차 마무리	배재	취부	용접1	용접2	사상/OUT
345	1420	B12DM	0501	166.80	213.45	0.00	0.00	1.15	4.20	6.52	6.80	4.45					03-5-30 8:00
345	1420	B12DM	0502	166.80	213.45	0.00	0.00	1.15	4.18	6.52	6.78	4.45					03-5-30 10:00
345	1413	D22SL	0531	296.20	268.60	0.00	0.00	1.19	6.16	11.22	3.79	0.38					03-5-30 10:00
345	1413	D22SL	0532	296.20	268.60	0.00	0.00	1.19	6.14	11.22	3.79	0.38					03-5-30 13:00
345	1413	D22SL	0533	127.20	163.29	0.00	0.00	0.74	3.65	7.81	0.75	5.98					03-5-30 15:00
345	1413	D22SL	0555	125.20	160.71	0.00	0.00	0.72	3.59	7.69	0.73	5.88					03-5-30 15:00
345	1398	B15D	0533	237.90	243.73	0.00	0.00	1.13	3.61	5.78	5.69	0.94					03-5-31 8:00
345	1398	B15D	0534	172.50	267.45	0.00	0.00	1.93	5.59	6.59	5.03	4.30					03-5-31 10:00

Fig. 7.10 Schedule chart

스케줄표로에서 스키드 이름을 확인 한 후 BOM으로부터 제품에 대한 정보를 얻어 스키드를 구성한다.(Fig. 7.11)

id_no	element	spart	qty_p	qty_s	sort	end_gb
001	F11	1P	3	0	0001	Y
001	F11	2	6	0	0006	Y
001	F11	3	3	3	0011	Y
001	F14	1P	1	0	0016	Y
001	F14	2	2	0	0021	Y
001	F14	3	1	1	0026	Y
001	F21	1P	4	0	0031	Y
001	F21	2	8	0	0036	Y
001	F21	3	4	4	0041	Y
001	F210	1P	1	0	0046	Y
001	F210	2	2	0	0051	Y
001	F210	3	1	1	0056	Y
001	F22	1P	1	0	0061	Y
001	F22	2	2	0	0066	Y
001	F22	3	1	1	0071	Y
001	F23	1P	1	0	0076	Y
001	F23	2	2	0	0081	Y
001	F23	3	1	1	0086	Y
001	F24	1P	1	0	0091	Y
001	F24	2	2	0	0096	Y
001	F24	3	1	1	0101	Y
001	G23	11	2	2	0106	Y
001	G23	12	1	1	0111	Y
001	G23	1P	1	0	0116	Y
001	G23	2A	2	2	0121	Y
001	G231	2A	2	2	0126	Y

Fig. 7.11 BOM (Bills Of Material)

## 7.5 시뮬레이션 결과

### (1) 작업 배분에 따른 결과

다음은 3차원 시뮬레이션 결과이다. 먼저 균등배분의 효과를 검증하기 위해 두 가지 경우를 비교하였다. 작업 영역을 로봇용접장 길이 방향(소조립 공장 라인 방향)으로 3등분하여 나눈 경우(Table 7.3, Three equal spaces)와 균등 배분 알고리즘을 적용한 경우(Table 7.3, Even distribution)이다. 세 번째와 다섯 번째 열은 작업 시간이 가장 긴 것과 짧은 gantry-robot의 cycle time의 차이이다. Table 7.3에서 보듯이 작업 시간 차가 크게 줄어든 것을 확인할 수 있다. 작업량의 적절한 배분을 통해 작업 시간 8.4분, cycle time의 10.0% 를 줄이는 효과를 볼 수 있다.

Table 7.3 Comparison of cycle time for distribution

unit : sec

SKID	Three equal spaces		Even distribution	
small amount of work (pieces deploy in one line)	5621.90	2632.05	5199.23	68.72
	6114.91		5157.53	
	3482.86		5226.25	
large amount of work (pieces deploy in two lines)	8187.94	3527.71	6986.39	336.25
	7584.71		7002.76	
	4660.23		7322.64	
	7535.06	2329.13	6820.71	161.85
7220.66	6982.56			
	5205.93		6962.04	
large amount of work (pieces deploy in two lines)	8730.69	1070.51	8730.68	145.91
	8992.97		8542.31	
	7922.46		8584.77	
	4123.51	4151.83	3179.54	610.20
	4151.83		3499.99	
	0.00		3789.74	
large amount of work (pieces deploy in two lines)	2169.24	945.21	2049.17	132.95
	2313.57		1916.22	
	1359.36		1918.79	
	2200.33	1243.57	2002.17	146.95
	2341.34		1976.54	
	1097.77		1855.22	

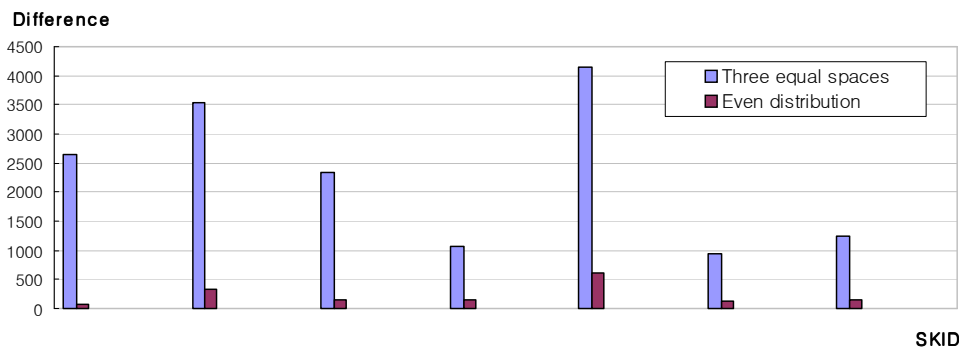


Fig. 7.12 Graph of difference for working time

**(2) 작업 경로에 따른 결과**

두 번째 검증은 GA를 이용한 작업 순서가 cycle time에 미치는 결과이다. 이를 위해 두 가지 모델을 비교하였다. 하나는 단순히 flat bar가 놓인 순서대로 작업을 하는 경우(Table 7.4, Arranged seq.)이다. 이것은 flat bar의 중심점을 기준으로 컨베이어 위에 놓여있는 순서대로 작업을 하는 것이다. 다른 하나는 작업 순서를 결정하는데 있어 GA를 적용한 것이다 (Table 7.4, Using GA). Table 7.4의 Reduction은 두 경우 작업 시간을 비교한 것이며, 작업 시간이 대부분 감소하였으며 일부는 증가하기도 하였다. 작업 경로의 최적화를 통해 작업 시간 12.4분, cycle time의 15.0%를 줄이는 효과를 볼 수 있었다.

Table 7.4 Comparison of cycle time according to change of working sequence using GA

unit: sec

SKID	Arranged seq.	Using GA	Reduction
small amount of work (pieces deploy in one line)	5199.23	4822.68	-376.55
	5157.53	5064.83	-92.70
	5226.25	4860.69	-365.56
large amount of work (pieces deploy in two lines)	6986.39	6713.67	-272.72
	7002.76	6617.49	-385.27
	7322.64	6670.65	-651.99
	6820.71	6623.80	-196.91
	6982.56	6708.57	-273.99
	6962.04	7018.55	56.51
large amount of work (pieces deploy in two lines)	8730.68	8438.03	-292.65
	8542.31	8229.98	-312.33
	8584.77	8522.05	-62.72
	3179.54	2994.48	-185.06
	3499.99	3399.44	-100.55
	3789.74	3336.94	-452.80
large amount of work (pieces deploy in two lines)	2049.17	1926.67	-122.50
	1916.22	1811.16	-105.06
	1918.79	1919.79	-1.00
	2002.17	1719.66	-282.51
	1976.54	1800.40	-176.14
	1855.22	1606.25	-248.97



## 제 8 장 결 론

소조립 로봇 용접 공정의 공정 최적화에 관한 본 연구에서 아래와 같은 결론을 얻었다.

- 1) 작업 계획 및 시뮬레이션 과정에서 소조립 공정 스킴 패턴의 필요성을 확인하였다.
- 2) 작업 계획 과정에서 유전자 알고리즘을 사용하여 작업 순서를 찾을 수 있는 방법을 찾아보았고, 시뮬레이션을 통해 그 결과를 검증하였다.
- 3) 임의의 물량에 대하여 cycle time을 추정할 수 있는 소조립 로봇 용접 공정의 3차원 시뮬레이터를 구현하였고, 이를 작업 계획에 반영할 수 있는 가능성을 제시하였다.

## 참 고 문 헌

- [1] 김여근, 윤복식, 이상복, “메타 휴리스틱”
- [2] Averill M. Law W. David Kelton, "Simulation Modeling and Analysis"
- [3] 하광철(2003), “XML 기반의 가공 생산 정보 정규화”
- [4] 권영탁, “유전자알고리즘을 이용한 TSP”, 충북대학교 대학원 석사학위논문
- [5] D.N.L. Levy, D.F. Beal., “*Heuristic programming in artificial intelligence*”
- [7] Thomas E. Morton and David W. Pentico. “*Heuristic scheduling systems : with applications to production systems and project management*”
- [8] 이말레, “근접 최적화 스케줄링을 위한 변형 유전자 알고리즘“, 중앙대학교 대학원 박사학위 논문
- [9] Chang-Sei Kim, Keum-Shik Hong, and Hans Yong-Sub Han, “PC-Based Off-Line Programming in Shipbuilding Industry: Open Architecture”

## 감사의 글

벌써 대학원 생활을 마감할 때가 왔습니다. 2년이란 시간에 짧게도 길게도 느껴졌었는데 논문을 내면서 지난 시절을 정리하자니 아쉬운 맘이 가장 큼니다. 대학원 생활을 하면서 저 자신에 대해서 좀 더 많이 알게 되었고, 이런저런 일들을 겪으면서 마음도 커진 것 같습니다.

많이 부족한 논문이지만 그래도 이만큼 올 수 있었던 것은 제가 가장 사랑하는 부모님이 계셨기 때문이라 생각합니다. 제게 건강한 심신을 주시고, 언제나 절 믿어주신 부모님께 감사합니다. 제 곁에 계신 걸 행복하게 생각하고 있습니다. 세상 누구보다도 사랑합니다.

3년 반동안 사랑으로 돌봐주신 박주용 교수님. 많이 표현하지 못했지만 교수님에게서 따뜻함을 많이 느꼈습니다. 감사드립니다. 자랑스런 제자가 되도록 꾸준히 노력하겠습니다.

그리고 제 논문을 심사 해주신 박석주 교수님과 조효제 교수님께 감사드립니다. 그동안 저와 함께 생활했던 연구실 식구들. 김세환 부장님, 최우현 부장님, 태인형, 재용형, 동환형, 재현형, 창섭, 현주, 해민. 고맙습니다.

대학 시절 만난 제 절친한 친구 진숙이와 은진이. 그리고 정신적으로 서로에게 큰 힘이 되고 있는 친구 인철이. 그리고 나의 잘생긴 남자친구 인중이! 많이 놀아주지 못해서 미안하고 이제 재미있게 놀자!! 대학 시절 서로 사랑하고 함께 했던 동기들. 고맙다. 언급하지 않으면 토라질 것 같아 호명하자면, 병성이형, 필용이, 길보, 호천, 묘정이, 경태. 그리고 졸업논문에 꼭 이름을 올려달라고 하던 인순이형, 절 많이 예뻐해 준 96학번 형들 고맙습니다. 그리고 최근 제게 큰 가르침을 주신 72년생 친구에게도 감사드립니다.