한국정밀공학회지 제 31 권 8 호 pp. 689-696
J. Korean Soc. Precis. Eng., Vol. 31, No. 8, pp. 689-696
ISSN 1225-9071(Print), ISSN 2287-8769(Online)

http://dx.doi.org/10.7736/KSPE.2014.31.8.689

# 스케줄링 문제를 위한 멀티로봇 위치 기반 다목적 유전 알고리즘

# Multi-Objective Genetic Algorithm based on Multi-Robot Positions for Scheduling Problems

최종훈 [1], 김제석 [1], 정진한 [1], 김정민 [2], 박장현 [3,✉]
Jong Hoon Choi[1], Je Seok Kim[1], Jin Han Jeong[1], Jung Min Kim[2], and Jahng Hyon Park[3,✉]

1 한양대학교 자동차공학과 (Department of Automotive Engineering, Hanyang Univ.)
2 아주대학교 기계공학과 (Department of Mechanical Engineering, Ajou Univ.)
3 한양대학교 미래자동차공학과 (Department of Automotive Engineering, Hanyang Univ.)
✉ Corresponding author: jpark@hanyang.ac.kr, Tel: +82-2-2220-0448

Manuscript received: 2014.5.2 / Revised: 2014.6.3 / Accepted: 2014.6.16

*This paper presents a scheduling problem for a high-density robotic workcell using multi-objective genetic algorithm. We propose a new algorithm based on NSGA-II(Non-dominated Sorting Algorithm-II) which is the most popular algorithm to solve multi-objective optimization problems. To solve the problem efficiently, the proposed algorithm divides the problem into two processes: clustering and scheduling. In clustering process, we focus on multi-robot positions because they are fixed in manufacturing system and have a great effect on task distribution. We test the algorithm by changing multi-robot positions and compare it to previous work. Test results shows that the proposed algorithm is effective under various conditions.*

Key Words: Scheduling Problem (스케줄링 문제), High-Density Robot (고밀도 로봇), Multi-Objective Genetic Algorithm (다목적 유전 알고리즘), Multi-Robot Position (멀티로봇 위치)

## 1. 서론

최근 3D 프린터가 각광받고 있는 사실에서 알 수 있듯이, 다양해진 소비자층의 이목을 끌고 소비자 개개인의 취향에 맞는 제품을 생산하려면 제품 공정에 다품종 소량생산 방식을 적용해야 한다. 이에 생산 시스템은 여러 대의 로봇이 공간적으로 떨어지도록 한 컨베이어 방식으로부터 좁은 공간에 고밀도로 배치된 다수의 로봇이 작업을 분담하여 수행하는 셀 생산 방식으로 변경되었다. 이 같이 고밀도로 매우 협소한 공간 내에 입체적으로 로봇을 배치하여 제품을 제조하는 공정은 기존의 공정보다 공장 면적 및 로봇 작업 수행 시간을 혁신적으로 줄일 수 있다.[1] 이러한 산업 트렌드에 맞춰 로봇에 작업을 할당하고 작업순서를 최적화하는 문제, 즉 멀티 로봇 스케줄링 문제를 해결해야 한다. 멀티 로봇 스케줄링을 풀기 위한 선행 연구들을 살펴보면, K-means Algorithm으로 작업량을 할

Copyright Ⓒ The Korean Society for Precision Engineering
This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

당하고 Genetic Algorithm, Simulated Annealing, Multiple Travel Salesman Problem과 같은 진화 알고리즘(Evolution Algorithm)으로 작업 순서를 스케줄링 하는 방법[2-5]과 작업량 할당 및 스케줄링을 동시에 수행하는 Genetic K-means Algorithm(GKA)방법 등이 있다.[6] 또한 스케줄링 기법에서는 지역개선을 위해 메타 휴리스틱(Meta-Heuristic)기법들을 적용하여 전체적인 스케줄링을 개선하는 방법도 많이 연구되어왔다. 하지만 K-means Algorithm은 초기 데이터에 영향을 많이 받고 로봇의 위치를 고려하지 않기 때문에 작업량을 최적으로 분배하기 어렵다. 작업 분배가 최적으로 이루어지지 않을 경우 멀티 로봇 스케줄링 기법의 전역 최적해(global optimum)를 찾기 어려운 단점이 존재한다. 또한 GKA와 같이 유전 알고리즘을 사용할 경우, 지역 최적해(local minimum)에 빠지지 않게 해주는 장점이 있지만 성능을 높이기 위해 population을 늘이면 연산량이 많아져 연산속도가 늦어진다. 본 논문에서는 이러한 문제를 해결하고자 멀티 로봇 스케줄링 문제에 자주 쓰이는 다목적 유전 알고리즘을 바탕으로 새로운 알고리즘을 제안한다. 제안한 알고리즘은 K-means Algorithm과 달리 초기 데이터에 영향을 받지 않고, 로봇의 위치에 따라 작업량이 클러스터링 되는 장점이 있다. 작업 공정에서 로봇의 대수와 위치는 고정되어있는 시설이기 때문에 그것을 고려하여 각 로봇에 작업량을 할당하고, 지역개선 과정을 수행하여 각 로봇에 대해 최적의 경로를 생성하는 것은 중요하다. 그리고 또한 작업량을 할당하는 클러스터링 과정과, 최적의 경로를 생성하는 지역개선 변이 연산 과정의 모집단 크기(population)를 다르게 하여 기존의 유전알고리즘보다 연산량을 줄여 효율을 높이고자 했다.

## 2. 문제 정의 및 가정

멀티 로봇 스케줄링 최적화 문제는 좁은 공간에 고밀도로 배치된 로봇들에게 작업을 분배하고 작업 순서를 최적화하는 문제로 정의한다. 이 문제는 잘 알려진 다수 순회 판매원 문제(Multiple Traveling Salesman Problem, MTSP)를 로봇으로 확장한 개념으로 볼 수 있다. 다수의 판매원이 동선을 낭비하지 않고 최대한 효율적으로 모든 도시를 방문해야 하듯이 다수의 로봇이 효율적이고 빠르게 작업을 수행하는 것이 필요하다. 본 논문은 정해진 위치에 배치되어있는 M 대의 로봇이 N 개의 용접점(작업점)을 분담하여 동시에 용접하는 상황에 대해서 연구하였다.

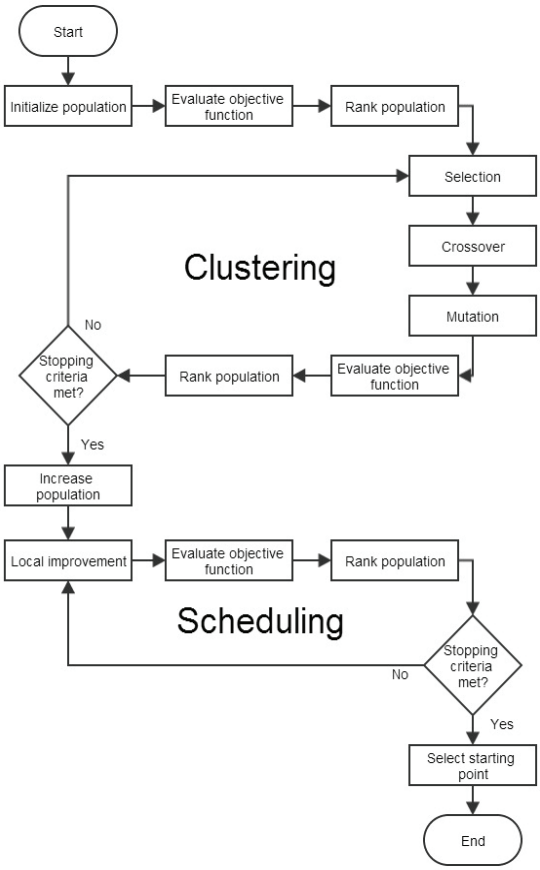



Fig. 1 Flow chart of multi-robot scheduling technique

## 3. 다목적 유전 알고리즘

### 3.1 소개

기존의 유전 알고리즘은 하나의 목적에 대해서만 전역 최적해(global optimum)을 찾는데 반해, 다목적 유전 알고리즘은 다중의 목적함수에 대해 최적화가 가능하다. 이 중에서도 NSGA-II (Non-Dominated Sorting Genetic Algorithm II)는 뛰어난 성능으로 다목적 최적화 문제에서 가장 많이 사용되고 있는 알고리즘 중 하나이다.[7] NSGA-II는 일반적인 유전 알고리즘과 거의 동일하지만 파레토 순위와 과밀 거리를 이용하여 각 염색체를 평가하는 방법을 사용한다. 파레토 순위는 하나의 염색체가 다수의 목적을 얼마나 만족하는지를 나타내며, 과

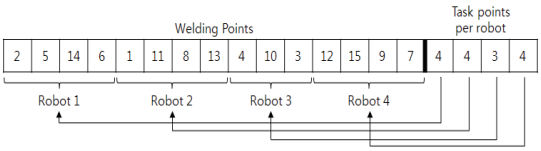



Fig. 2 Two-part chromosome representation

밀 거리는 염색체가 모집단에서 얼마나 유일한지를 나타내 주는 지표가 된다. 이와 같은 파레토 순위와 과밀 거리의 개념으로 각 유전자들의 상대적 우월성을 평가하고, 이 결과를 바탕으로 다음 세대에 선택될 염색체를 선택하게 된다.[8] 본 논문에서는 이러한 특성을 가지는 NSGA-II를 이용하여 멀티 로봇 스케줄링 최적화 문제를 Fig. 1과 같이 구성하고, 다음과 같이 제안한다.

### 3.2 Initialize population

N 개의 용접점과 M 대의 로봇에 대해서, 작업을 분배하고 작업 순서를 결정하기 위해 임의의 염색체를 생성하고 모집단으로 활용한다. 염색체 표현 방법으로는 MTSP에서 뛰어난 성능으로 많이 쓰이는 이분 염색체 표현법(Two-part chromosome representation, TCR)을 사용하였다.[4] Fig. 2에서 첫 번째 부분은 용접점을 첫 번째 로봇의 용접 시작점부터 마지막 로봇의 마지막 용접점까지 순서대로 나열한 것이고, 두 번째 부분은 각 로봇에 할당된 용접점의 수를 표현한다.

### 3.3 Multi-objective Function

#### 3.3.1 최대 작업 시간

멀티 로봇 스케줄링에서 최적해는 기본적으로 최대 작업 시간이 최소가 되도록 하여야 한다. M 개의 로봇과 N개의 용접점이 주어졌을 때, 로봇 m에 할당된 $N_m$ 개의 용접점에 대해 작업 시간 $t_m$ 은 다음과 같이 주어진다. 여기서, $X_{m,i}$ 는 로봇 m의 i번째 용접점, $\dot{X}$ 은 로봇 end effector의 이동속도, $t_k$ 는 용접점에 대한 작업 시간 상수이다.

$$\begin{aligned} t_{m,1} &= \sum_{i=2}^{N_m}\left(\frac{\left|X_{m,i}-X_{m,(i-1)}\right|}{\dot{X}}\right)+N_m t_k \\ t_{m,2} &= \frac{\left|X_{m,1}-X_{m,\mathrm{N}_m}\right|}{\dot{X}} \\ t_m &= t_{m,1}+t_{m,2} \end{aligned} \tag{1}$$

따라서, 각각의 로봇이 작업을 수행한 시간 중 최대 작업 시간을 최대값으로 선택하여 아래의 목적함수를 최소화시키고자 한다.

$$F_1 = \max(t_1,\cdots,t_M) \tag{2}$$

#### 3.3.2 모든 로봇의 작업 시간 총합

최대 작업 시간만을 목적함수로 정하는 경우 최대 작업 시간을 가지는 로봇에만 경로가 최적화되고 나머지 로봇에는 최적의 경로가 생성되지 않은 경우가 발생하였다. 이에 모든 로봇의 작업 시간 총합을 목적함수로 두어 작업의 효율성과 용접점의 고른 분배를 수치로 판단할 수 있게 하였다.

$$F_2 = \sum_{i=1}^{M} t_i \tag{3}$$

### 3.4 NSGA-II

NSGA-II(Non-dominated Sorting Genetic Algorithm)는 일반적인 유전 알고리즘과 동일한 연산과정을 거친다. 임의의 타당한 해로 구성된 초기 모집단(population)을 가지고, 이 모집단에서 특정 확률 분포에 근거하여 부모가 되는 두 원소를 선택(selection)한다. 이 둘은 교차(crossover) 연산자를 통해 결합되어 자식을 낳게 된다. 자식은 변이(mutation) 연산자에 의해 변형되는데, 경험하지 못한 탐색공간을 모집단에 도입시키기 위해서이다. 그 다음에 자식에 대해 지역 개선(local improvement) 과정을 수행한다. 이 과정은 종료 조건을 만족할 때까지 반복적으로 진행하여 진화를 거듭한다. 이러한 과정은 비슷하지만 NSGA-II는 일반적인 유전 알고리즘과 다르게 염색체를 평가하는 방법으로 파레토 순위와 과밀 거리를 이용하여 다른 유전알고리즘 보다 더 나은 해를 찾을 수 있고, 더 나은 수렴성을 나타낸다.

본 논문에서는 NSGA-II 단계에서 클러스터링과 스케줄링을 구분하여 수행하였다. 선택, 교차, 변이 연산자를 통해 각 로봇에 용접점을 분배하는 클러스터링을 수행하고, 모집단의 크기를 증가시킨 후 지역 개선을 통해 각 클러스터의 스케줄링을 수행하였다.

#### 3.4.1 Rank

염색체를 평가하는 방법으로 파레토 순위와 과밀 거리를 이용한다. 파레토 순위는 하나의 염색체가 두 개의 목적을 얼마나 만족하는지를 나타내

며, 만약 파레토 순위가 높은 경우 대상 염색체는 두 목적을 매우 만족하고 있음을 의미한다. 다음은 파레토 순위를 결정하는 과정을 나타낸다.

① 파레토 순위를 나타내는 변수 r을 1로 설정한다.

② 모집단에서 파레토 순위가 정해지지 않은 염색체에 대한 파레토 경계를 구한다.

③ 파레토 경계인 모든 염색체에 파레토 순위 r을 할당한다.

④ r을 1만큼 증가시킨다.

⑤ 만약 모든 염색체에 파레토 순위가 할당되면 종료하고, 그렇지 않으면 ②로 이동한다.

과밀 거리는 염색체가 모집단에서 얼마나 유일한지를 나타내주는 지표가 된다. 만약 한 염색체가 모집단에서 유일성이 높다면 과밀 거리의 값이 커지고, 반대로 염색체가 모집단에서 유일성이 낮다면, 과밀 거리의 값이 작아진다. 과밀 거리는 다음과 같은 식으로 정의된다.

$$CD_i = \sum CD_{im} \tag{4}$$

위 수식에서, $CD_i (1 \le i \le l)$ 는 염색체 i에 대한 과밀 거리를 나타내고, $CD_{im}$ 은 목적 함수 m에 대한 과밀 거리 값을 나타낸다. 그리고 $l$ 은 같은 파레토 순위에 있는 염색체 집단의 크기를 나타낸다. $CD_{im}$ 은 아래의 식과 같다.

$$CD_{im} = \begin{cases} \dfrac{f_m(x_{i+1}) - f_m(x_{i-1})}{f_m(x_{\max}) - f_m(x_{\min})} & \text{for i} = 2, \cdots, l-1 \\ \infty & \text{for i} = 1 \text{ or } l \end{cases} \tag{5}$$

이때 $CD_{im}$ 은 모집단 중 i번째 염색체의 밀도와 연관되어 있다. 예를 들어 i번째 염색체 주변의 밀도가 낮아진다면 과밀 거리의 값은 더욱 커지게 된다. 즉, 만약 염색체의 과밀 거리 값이 매우 크면 해당 염색체와 비슷한 비용을 가진 염색체의 수가 매우 적다는 것을 의미한다. 선택 과정 동안 파레토 순위를 사용함으로써 전체적인 해는 결국 더 높은 적합도를 향해 수렴하게 된다. 그리고 높은 과밀 거리를 가진 염색체를 선택함으로써 해들의 다양성이 보장될 수 있다.

### 3.4.2 Selection

NSGA-II는 다목적 함수에 따른 파레토 순위와 과밀거리를 구하고, 이 두 가지 인자를 이용하여

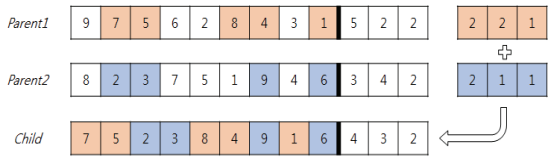



Fig. 3 Two-part chromosome crossover

Tournament Selection에서 P/2 개의 개체를 선정한다. Tournament Selection에서 선택된 P/2 개의 개체에 대해서 교차연산과 변이연산을 각각 확률적으로 수행하여 P 개의 개체를 생성한다.[9]

### 3.4.3 Crossover

멀티 로봇 스케줄링에서 최적해를 탐색하는데 있어 탐색 성능을 개선하기 위해 Fig. 3과 같은 이분 염색체 교배(two-part chromosome crossover, TCX) 연산을 도입하였다.[10] 다음은 TCX 연산을 수행하는 과정을 나타낸다.

① 모집단에서 특정 확률 분포에 근거하여 부모가 되는 두 원소를 선택한다.

② 부모1에서 각 로봇에 대해 임의로 일정 범위를 선택한다.

③ 부모1에서 선택되지 않은 유전자들을 부모2에서 선택한다.

④ ②와 ③에서 선택된 유전자들을 조합한다.

⑤ 새로운 자식의 two-part chromosome을 생성한다.

### 3.4.4 Mutation

본 논문에서 제안하는 멀티 로봇 스케줄링에서의 변이 연산은 기존의 변이 연산과는 달리 지역 최적화와 유사한 방식으로 수행된다. 그 방식은 다음과 같다.

① 모든 용접점을 용접점과 가장 가까운 로봇으로 배정한다.

② 랜덤으로 클러스터(로봇)를 선택하여 배정된 용접점 중 로봇과 가장 멀리 떨어진 점을 그 점과 두 번째로 가까운 로봇으로 옮긴다.

③ 랜덤으로 클러스터(로봇)를 선택하여 배정된 용접점 중 클러스터의 centroid와 가장 멀리 떨어진 점을 그 점과 두 번째로 가까운 centroid를 가지는 클러스터로 옮긴다.

④ 랜덤으로 클러스터(로봇)를 선택하여 배정된 용접점 중 로봇과 x, y, z축으로 가장 멀리 떨어진 점을 그 점과 두 번째로 가까운 로봇으로 옮긴다.

⑤ 랜덤으로 클러스터(로봇)를 선택하여 배정된 용접점 중 클러스터의 centroid와 x, y, z축으로 가장 멀리 떨어진 점을 그 점과 두 번째로 가까운 centroid를 가지는 클러스터로 옮긴다.

⑥ 위의 변이연산 수행 후, 각 클러스터에 대하여 가까운 용접점을 순차적으로 순회 하게끔 용접순서를 재배치한다.

각 변이 연산은 일정한 확률에 따라 일어나므로 한 염색체에 적용되는 것이 아니라 병렬적으로 각각 다른 염색체에서 동시다발적으로 수행된다. 위 변이연산을 수행하면 로봇위치를 기준으로 클러스터링이 수행되고 로봇과 제일 멀리 떨어진 점, 또는 클러스터의 centroid와 가장 멀리 떨어진 점을 근처의 로봇에게 넘겨주므로 클러스터링이 균일하게 되는 효과가 있다. ⑥은 염색체를 평가할 때 클러스터 내에서 용접 순서가 꼬인 경우 목적함수 값이 높게 나오는데 이를 배제하고 클러스터링이 얼마나 잘되었는지 판단하기 위해서 수행하였다.

### 3.4.5 Increase population

선택, 교차, 변이 연산자를 통해 각 로봇에 용접점을 분배하는 클러스터링을 수행하고, 모집단의 크기를 증가시킨 후 지역 개선을 통해 각 클러스터의 스케줄링을 수행하였다. 모집단의 크기를 바꾼 이유는, 클러스터링 단계에서는 소수의 우수한 염색체에 옮기고 싶은 점을 옮기는 연산을 적용, 즉 방향성을 가진 변이 연산을 반복 적용하기 위함이고, 스케줄링 단계에서는 오로지 확률에 기반을 둔 지역 개선 과정을 수행하기 때문에 모집단의 크기를 증가시켜 다수의 염색체에 적용하기 위함이다.

### 3.4.6 Local improvement

클러스터링이 완료된 염색체에 대해서 스케줄링상 꼬인 부분을 해결하기 위해 다음과 같은 지역개선 과정을 클러스터 내에서 수행한다.

① 연속된 두 점을 선택하여 순서를 뒤집는다.

② 랜덤으로 두 점을 선택하여 순서를 교환한다.

③ 랜덤으로 두 점을 선택하여 그 사이에 있는 점들의 순서를 뒤집는다.

④ 랜덤으로 한 부분을 선택하여 순서를 옮긴다.

위 과정을 Fig. 4에 나타내었다.

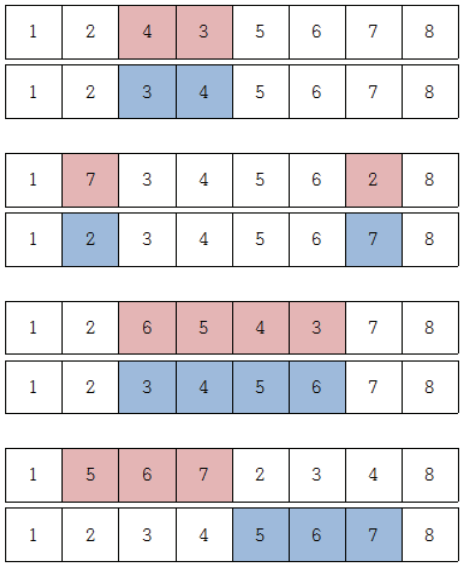



Fig. 4 Local improvement

### 3.4.7 Select starting point

클러스터링과 스케줄링이 완료된 후, 각 클러스터마다 각 로봇의 초기 위치와 가장 가까운 용접점에서부터 순회작업을 시작하도록 지정해준다.

## 4. 시뮬레이션

시뮬레이션 환경은 자동차 제조 공정에서 산업용 로봇을 고밀도로 배치하여 차량을 용접하는 상황을 가정한다. 차량 프레임을 따라 위치한 336개의 용접점을 각 로봇에 분배하고 순회경로를 생성하여 총 작업시간을 최소화하고자 한다. 로봇의 이동속도는 500 mm/s, 용접시간은 5 s/spot으로 설정하였다. 클러스터링 단계와 스케줄링 단계의 population은 10에서 50으로 5배 커지게 하였다. 로봇의 대수와 위치에 따라 적합한 해를 얻을 수 있는 지 확인하기 위해 다음과 같은 시나리오에 따라 성능을 확인하였다.

첫 번째는 6대의 로봇이 위아래로 3대씩 위치한 경우이다. 클러스터링과 스케줄링 각 단계에서 1000, 5000번의 반복횟수 동안 해가 좋아지지 않으면 알고리즘을 종료하였다. 총 5회의 실험 결과는 Table 1과 같다.

두 번째는 8대의 로봇이 위아래로 4대씩 위치한 경우이다. 클러스터링과 스케줄링 각 단계에서 2000, 10000번의 반복횟수 동안 해가 좋아지지 않으면 알고리즘을 종료하였다. 총 3회의 실험 결과는 Table 2와 같다.

Table 1 Simulation result of 6 robots

| No. | F1 | F2 | Task distribution |
|---|---|---|---|
| 1 | 280.661 s | 1682.97 s | 56/56/56/56/56/56 |
| 2 | 280.668 s | 1683.23 s | 56/56/56/56/56/56 |
| 3 | 280.631 s | 1683.02 s | 56/56/56/56/56/56 |
| 4 | 280.741 s | 1683.24 s | 56/56/56/56/56/56 |
| 5 | 280.777 s | 1683.15 s | 56/56/56/56/56/56 |

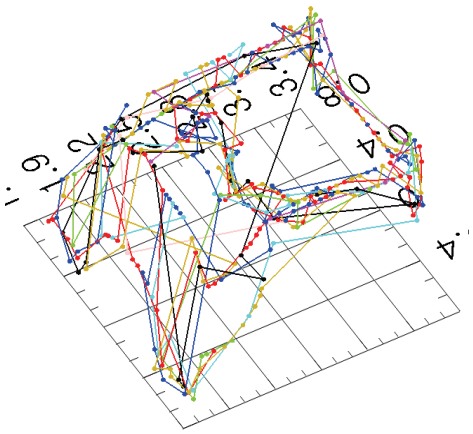


Fig. 5 Before scheduling

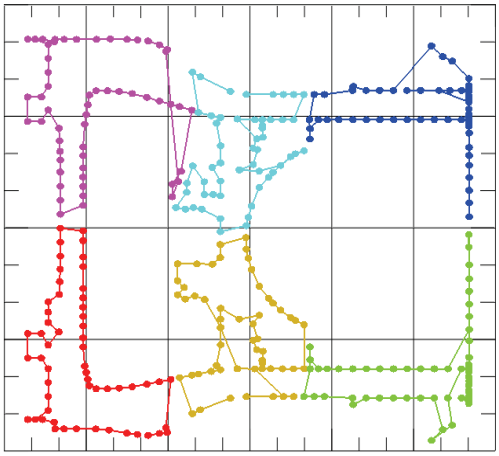


Fig. 6 Simulation result of 6 robots

Table 2 Simulation result of 8 robots (1)

| No. | F1 | F2 | Task distribution |
|---|---|---|---|
| 1 | 210.632 s | 1683.27 s | 42/42/42/42/ 42/42/42/42 |
| 2 | 210.521 s | 1683.02 s | 42/42/42/42/ 42/42/42/42 |
| 3 | 210.558 s | 1683.42 s | 42/42/42/42/ 42/42/42/42 |

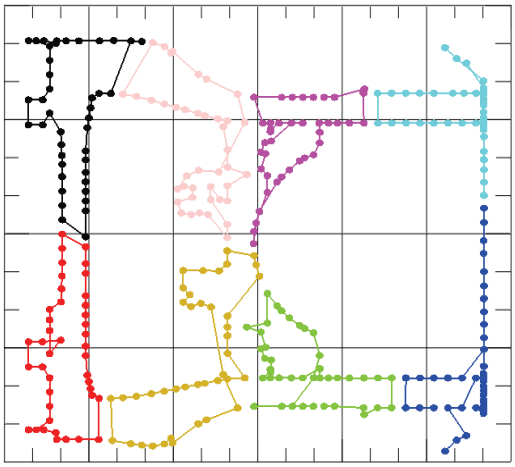


Fig. 7 Simulation result of 8 robots (1)

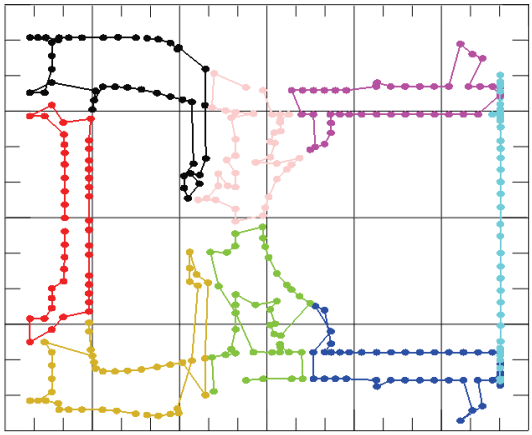
Table 3 Simulation result of 8 robots (2)

| No. | F1 | F2 | Task distribution |
|---|---|---|---|
| 1 | 210.988 s | 1684.15 s | 42/42/42/42/ 42/42/42/42 |
| 2 | 210.923 s | 1684.38 s | 42/42/42/42/ 42/42/42/42 |
| 3 | 210.805 s | 1683.68 s | 42/42/42/42/ 42/42/42/42 |

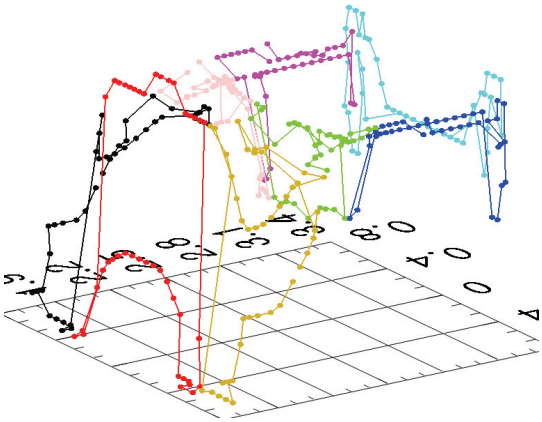
세 번째는 8대의 로봇이 위아래로 3대씩, 양옆으로 1대씩 위치한 경우이다. 클러스터링과 스케줄링 각 단계에서 2000, 10000번의 반복횟수 동안 해가 좋아지지 않으면 알고리즘을 종료하였다. 총 3회의 실험 결과는 Table 3과 같다.

시뮬레이션 결과를 보면, 로봇의 대수와 위치에 따라 용접점의 분배가 고르게 이루어졌음을 확인할 수 있다. 또한 목적함수인 최대작업시간과 작업시간 총합도 실험을 여러 번 해도 거의 비슷하게 나오는 것으로 보아 초기데이터에 영향을 받지 않는 것을 알 수 있다.

알고리즘의 우수성을 입증하기 위해 기존의 알고리즘인 GKA(Genetic K-means Algorithm)+LK(Lin-Kernighan)로 동일한 작업점을 로봇 6대에 분배하는 시뮬레이션을 수행하였다. GKA+LK는 초기 단계에서 k-means 알고리즘을 이용하여 전체 로봇에 대해 작업을 분배하고, LK 알고리즘을 이용하여 각 로봇에 대해 최적의 경로를 생성하는 방법으로 제안한 알고리즘과 큰 틀은 비슷하다. 하지만 초기 데이터에 영향을 많이 받고 클러스터간의 작업량 할당이 적절하지 않은 문제가 있다. Table 4를



(a)



(b)

Fig. 8 Simulation result of 8 robots (2)

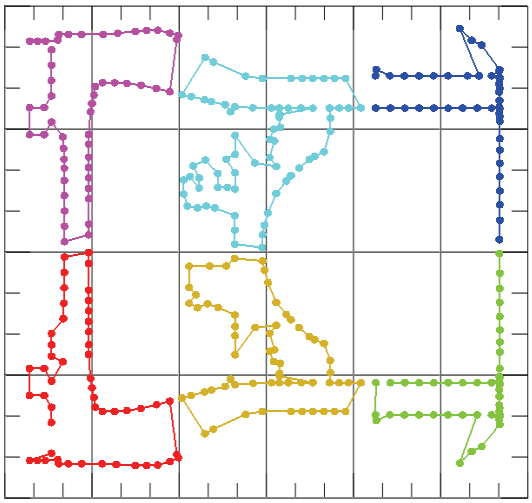


Fig. 9 Simulation result of GKA+LK

Table 4 Simulation result of GKA+LK and proposed algorithm

| Algorithm | F1 | F2 | Task distribution |
|---|---|---|---|
| GKA+LK | 375.442 s | 1687.54 s | 61/45/56/ 58/51/66 |
| Proposed Algorithm | 280.611 s | 1682.97 s | 56/56/56/ 56/56/56 |

보면 제안한 알고리즘이 기존 알고리즘에 비해 최대 작업시간도 많이 단축시키고 용접점도 고루 분포시켜 더 좋은 해에 도달하는 것을 알 수 있다.

## 5. 결론

본 연구는 좁은 공간에 고밀도로 배치된 다수의 로봇이 작업을 분담하여 수행하는 셀 생산 시스템의 최적화 문제에 새로운 접근을 시도하였다. 제조 공정 시설에서 로봇의 대수와 위치는 고정되어있으므로 이를 고려하여 작업을 분배하는 알고리즘을 제안하였다. 그리고 기존의 다목적 유전 알고리즘인 NSGA-II를 기반으로 클러스터링과 스케줄링을 구분 지어 알고리즘을 만들었고, 알고리즘을 효율적으로 수행하기 위해 클러스터링 단계보다 스케줄링 단계의 population을 증가시켰다. 그 결과, 기존의 k-means 알고리즘의 단점인 초기데이터에 영향을 받는 것과 클러스터의 개수를 결정하기 어려운 문제를 극복하였고 다른 유전 알고리즘보다 효율적인 작업 분배로 더 좋은 해에 도달하였다.

향후 연구로는 작업을 분배하고 순서를 정할 때 로봇과 로봇, 또는 로봇과 다른 구조물의 충돌을 고려하여 이를 회피하는 알고리즘에 대한 연구가 필요하다.

## 후 기

본 연구는 산업통상자원부 산업원천기술개발사업 (10033555, 고밀도 작업용 멀티로봇 공정 효율화 및 공통기술 개발)의 연구결과와 미래창조과학부 및 정보통신산업진흥원의 IT융합 고급인력과정 지원사업의 연구결과로 수행되었음(NIPA-2014-H0401-14-1001)

## REFERENCE

1. Hara, S. and Azuma, K., “Cell Production System for Assembly,” Robotics and Computer-Integrated

Manufacturing, Vol. 4, No. 3, pp. 379-385, 1988.
2. Deb, K., "Multi-Objective Optimization using Evolutionary Algorithms," John Wiley & Sons, 2001.
3. Bektas, T., "The Multiple Traveling Salesman Problem: An Overview of Formulations and Solution Procedures," Omega, Vol. 34, No. 3, pp. 209-219, 2006.
4. Carter, A. E. and Ragsdale, C. T., "A New Approach to Solving the Multiple Traveling Salesperson Problem using Genetic Algorithms," European Journal of Operational Research, Vol. 175, No. 1, pp. 246-257, 2006.
5. Konak, A., Coit, D. W., and Smith, A. E., "Multi-Objective Optimization using Genetic Algorithms: A Tutorial," Reliability Engineering & System Safety, Vol. 91, No. 9, pp. 992-1007, 2006.
6. Krishna, K. and Murty, M. N., "Genetic K-Means Algorithm," Proc. of the IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, Vol. 29, No. 3, pp. 433-439, 1999.
7. Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., "A Fast and Elitist Multiobjective Genetic Algorithm: Nsga-II," Proc. of the IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2, pp. 182-197, 2002.
8. Kim, K. H., Ku, T. W., and Baek, J. M., "Reliability Allocation Optimization for Software Modules using Multi-Objective Genetic Algorithm," Journal of Korea Information Science Society: Software and Applications, Vol. 39, No. 3, pp. 205-214, 2012.
9. Seshadri, A., "A Fast Elitist Multiobjective Genetic Algorithm: NSGA-II," MATLAB Central, pp. 1-4, 2006.
10. Yuan, S., Skinner, B., Huang, S., and Liu, D., "A New Crossover Approach for Solving the Multiple Travelling Salesmen Problem using Genetic Algorithms," European Journal of Operational Research, Vol. 228, No. 1, pp. 72-82, 2013.
11. Walshaw, C., "A Multilevel Lin-Kernighan-Helsgaun Algorithm for the Travelling Salesman Problem," Citeseer, pp. 1-9, 2001.
12. Helsgaun, K., "General *k*-Opt Submoves for the Lin-Kernighan TSP Heuristic," Mathematical Programming Computation, Vol. 1, No. 2-3, pp. 119-163, 2009.
13. Nallusamy, R., Duraiswamy, K., Dhanalaksmi, R., and Parthiban, P., "Optimization of Non-Linear Multiple Traveling Salesman Problem using K-Means Clustering, Shrink Wrap Algorithm and Meta-Heuristics," International Journal of Nonlinear Science, Vol. 9, No. 2, pp. 171-177, 2010.
14. Lim, S. S., Kim, J. S., and Park, J. H., "Multi-Objective Genetic Algorithm for High-Density Robotic Workcell," Proc. of the IEEE International Symposium on Robotics (ISR), pp. 1-3, 2013.