

FMS 스케줄링을 위한 Priority 함수의 자동 생성에 관한 연구

김창욱*, 신호섭*, 장성용**, 박진우*

* 서울대학교 산업공학과

** 서울산업대학교 산업공학과

ABSTRACT

Most of the past studies on FMS scheduling problems may be classified into two classes, namely off-line scheduling and on-line scheduling approach. The off-line scheduling methods are used mostly for FMS planning purposes and may not be useful for real time control of FMSs, because it generates solutions only after a relatively long period of time. The on-line scheduling methods are used extensively for dynamic real-time control of FMSs although the performance of on-line scheduling algorithms tends to vary dramatically depending on various configurations of FMS. Current study is about finding a better on-line scheduling rules for FMS operations.

In this study, we propose a method to create priority functions that can be used in setting relative priorities among jobs or machines in on-line scheduling. The priority functions reflect the configuration of FMS and the user-defined objective functions. The priority functions are generated from diverse dispatching rules which may be considered a special priority functions by themselves, and are used to determine the order of processing and transporting parts.

Overall system of our work consists of two modules, the Priority Function Evolution Module (PFEM) and the FMS Simulation Module (FMSSM). The PFEM generates new priority functions using input variables from a terminal set and primitive functions from a function set by genetic programming. And the FMSSM evaluates each priority function by a simulation methodology. Based on these evaluated values, the PFEM creates new priority functions by using crossover, mutation operation and probabilistic selection. These processes are iteratively applied until the termination criteria are satisfied. We considered various configurations and objective functions of FMSs in our study, and we seek a workable solution rather than an optimum or near optimum solution in scheduling FMS operations in real time. To verify the viability of our approach, experimental results of our model on real FMS are included.

Keywords : FMS, Scheduling, Priority Functions, Genetic programming, Simulation

1. 연구의 배경 및 목적

job shop처럼 적은 투자 비용으로 생산 제품을 빨리 변경할 수 있으면서도, flow shop처럼 생산 용량이 높은 부하 정도로 사용되어 질 때 제품 한 단위 당의 낮은 비용으로 높은 생산성을 취할 수 있는 제조 방식의 시스템으로 FMS (Flexible Manufacturing System : 유연 생산 시스템)가 나타나게 되었다.

FMS scheduling 문제를 풀기 위한 기존의 연구들로 여러 가지 접근 방식들로 크게 off-line scheduling 방법과 on-line scheduling 방법의 두 가지 접근 방식이 있으나 off-line 방식은 계산량의 과다와 실시간 대처에 대해서 전반적으로 부족한 측면을 갖고 있고, on-line 방식은 계산 방식이 단순하여 좋은 해를 지속적으로 구해내기 힘든 측면을 갖고 있다.

각자의 시스템마다 대상 시스템에 적합한 scheduler를 개발하는 것이 더 실제적인 의미가 있다. 그러나 실제로 자신의 시스템에 적합한 scheduler를 개발하는 과정은 개발자에게 많은 시스템의 성능에 관련된 많은 직관을 요구하며 많은 시행착오가 필요하다. 따라서 시스템의 특

성을 반영한 새로운 scheduler를 개발하는 방법을 자동화하는 체계적인 접근 방법이 필요하다고 할 수 있다. 이것이 바로 본 연구를 수행하게 된 배경이다.

앞에서 언급하였듯이 일반화 시킨 최적화 방법론을 생성하는 것은 거의 불가능해 보이기 때문에 priority function을 기반으로 하는 on-line scheduler를 대상으로 off-line에 자동으로 생성시킬 수 있도록 하는 프레임워크를 제안하고 적용시키는 것을 본 연구의 목적으로 삼는다.

연구 목적을 이루기 위한 방법론으로는 genetic programming이라고 하는 방법론을 사용하여 새로운 priority function을 탐색하고 발견할 수 있도록 할 것이며, 다양한 시스템들의 특성이 반영될 수 있도록 하는 과정은 적용 대상 시스템을 모델링한 simulation의 수행을 통해서 이루어지게 된다. 또한 이러한 탐색과 검증의 과정의 전체 순환 루프의 수행을 하나의 프레임 속에서 자동으로 수행될 수 있도록 하는 방법론을 제시한다.

2. 관련 연구 현황 및 분석

2.1. FMS scheduling 관련 연구 현황

기존의 FMS scheduling에 대한 연구들을 사용한 방법론에 의해 분류하면 크게 1)mathematical programming 접근 방법, 2)multi-criteria decision making 접근 방법, 3)heuristic을 이용한 방법,

4)control theory를 이용한 방법, 5)simulation을 이용한 방법, 6)AI를 이용한 방법 등으로 크게 나눌 수 있다. 기존의 연구들을 FMS의 모델링 시에 고려한 사항들의 관점에서 정리하면 다음의 [표1]과 같다.

[표 1] 기존 연구에서 고려한 FMS 요인들에 의한 분류

	Route flexibility	Tool Slots	Part Transport	Machine availability	Buffer spaces	Pallets
Chandra and Talvage(91)	○	×	×	×	×	×
Chen and Chung(91)	○	○	×	×	×	×
Wilson(89)	○	○	×	×	×	○
Ishii and Talvage(91)	×	×	○	×	×	×
Sauve and Collinot(87)	○	○	×	×	○	×
Choi and Malstrom(88)	○	×	×	○	×	○
Park(89)	○	×	×	×	○	×
Kusiak(86)	○	○	○	×	×	○
O'Grady and Lee(88)	○	○	○	×	○	×
Chang and Sullival(90)	○	×	○	×	×	×
Buzacott(82)	○	×	○	×	○	×
Slomp(88)	○	×	○	×	○	×
Akella(84)	×	×	○	○	×	×
Lee and Jung(89)	○	×	×	×	×	○
Wu and Wysk(89)	×	×	×	×	×	×
Sabunchoglu(92)	×	×	○	×	○	×
Han and McGinnis(89)	×	×	×	○	○	×

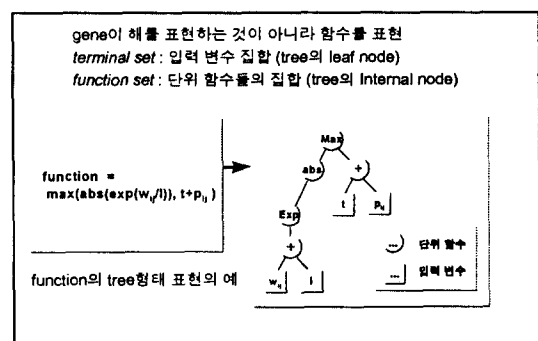
[표1]에서 보는 바와 같이 각 연구들이 바라보는 FMS에 대한 관점들이 매우 다양하며 또한 여러 가지 요인들을 한꺼번에 고려한 연구도 또한 없다.

이 밖에 또한 좋은 scheduling 방법을 고안하는 것을 시스템에서 가장 중요한 인자를 찾는 것으로 보고 새로운 인자를 찾는 노력을 수행한 연구들과 또한 기존의 dispatching 방법론들을 조합 시키기 위한 연구들이 최근에 많이 연구되고 있다. 그러한 연구들을 살펴보면 Dorndorf는 각 기계에서 사용할 dispatching rule들을 gene으로 표현하여 genetic algorithm을 수행하였다.

2.2. 학습 기법 관련 연구 현황

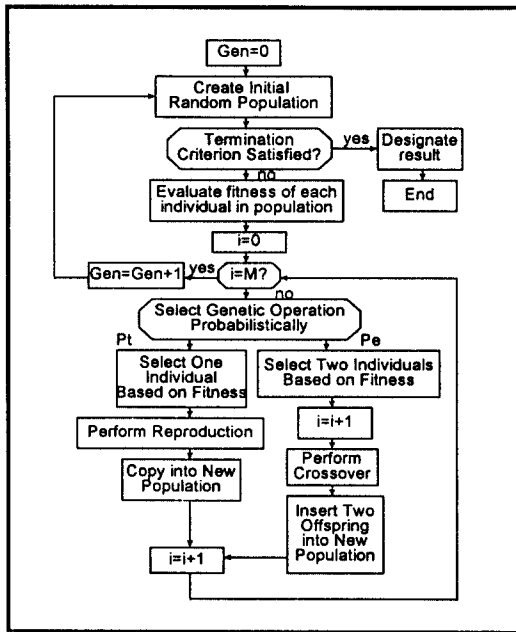
여러 가지 탐색 또는 학습과 관련된 기법들이 여러 가지 연구되어 왔다. 그러한 기법들로는 genetic programming, genetic algorithm, hill-climbing, neural network, decision tree, simulated annealing 등이 있다. 이러한 여러 가지 학습 기법들 중에서 본 연구에서 새로운 priority function을 학습하기 위한 방법으로 genetic programming을 사용한다.

이러한 genetic programming에서 function의 gene을 표현하는 것은 [그림 1]과 같다.



[그림 1] genetic programming에서의 function의 gene 표현

간략하게 genetic programming의 메카니즘을 살펴보면 [그림2]와 같다. [12]



[그림 2] genetic programming의 흐름도

3. FMS scheduler의 자동 생성 시스템

특정 시스템에 적합한 scheduling 방법을 개발하는 과정은 다음과 같다.

1. 새로운 scheduling 방법을 생성하고, (Generation)
2. 대상의 시스템에 적용하여 운영해보고, (Adaptation)
3. 적용해 본 scheduling 방법의 성능을 파악하고, (Evaluation)
4. 실험해 본 scheduling 방법들 중에서 나은 것을 선택한다. (Selection)

위의 과정은 만족할 만한 scheduling 방법을 찾을 때 까지 계속해서 수행해야 하는 일이다.

대상의 시스템에 적용하여 운영해 보는 부분에 대하여는 시뮬레이션을 수행하고, 새로운 scheduling 방법을 생성하는 과정을 priority function을 새로 생성하는 과정으로 하였다.

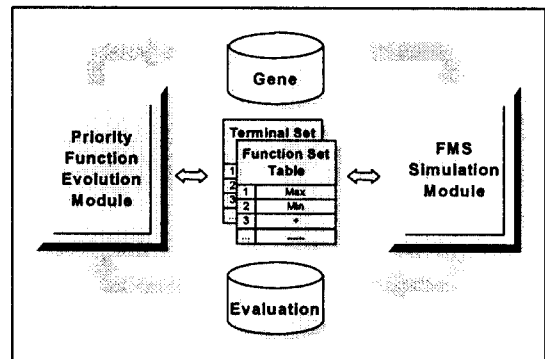
3.1. 전체 시스템의 구성

전체 시스템은 크게 Priority Function Evolution Module과 FMS Simulation Module의 두개의 모듈로 구성된다. Priority Function Evolution Module에서 생성된 결과인 priority 함수를 FMS Simulation Module에서 읽어 들이고, 해당 함수를 사용한 simulation을 수행하여 그 함수의 성능을 평가한

다. 그렇게 평가한 결과는 다시 Priority Function Evolution Module에서 다시 읽어 들이고, 다음 단계에서 새로운 함수를 생성시킬 때 좋은 특성을 갖는 함수들이 발전될 수 있도록 유도하여 또 다른 함수들을 생성시키는 과정을 거치게 된다. 이러한 일련의 과정들은 순환하면서 수행되다가 처음에 정한 시스템 종료 조건을 만족시키게 되면 종료하고, 이전까지 생성시키고 평가한 것들의 priority 함수들 중에서 가장 성능이 좋은 것을 제시해 준다.

또한 이러한 전체의 구성을 독립된 두개의 모듈로 구분함으로써 얻을 수 있는 이점들은 다음과 같다.

1. 다양한 FMS 모델의 수용 가능
2. 다양한 시스템 성능 평가 기준의 쉬운 적용
3. 다양한 FMS 모형의 수용 가능
4. 새로운 primitive function set과 terminal set의 수용 가능
5. 새로운 function evolution method의 수용 가능



[그림 3] 전체 시스템의 구성

3.2. FMS Simulation 모듈

전체 시스템의 두개의 모듈 중의 하나인 FMS Simulation Module에서는 사용자가 지정한 대상의 FMS 모델에 대하여 주어진 priority function들의 성능을 평가하게 된다.

3.2.1. FMS 운영 단계에서의 의사결정 문제

FMS를 운영하는 데에 있어서의 scheduling에 관한 의사결정 문제는 결국 각 entity에서의 sequencing 문제이다. 이러한 특성을 기반으로 FMS에서의 각 entity입장에서의 의사결정은 크게 네 가지로 분류할 수 있다.

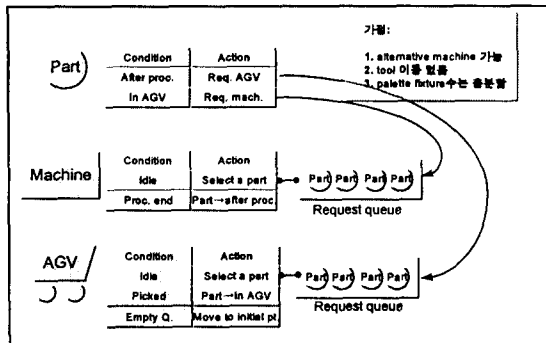
1. 부품이 다음 단위공정을 수행할 기계를 선택하는 문제

2. 기계가 대기 부품들 중에서 다음에 가공을 수행할 부품을 선택하는 문제
3. 부품이 운반할 자재 운반 시스템을 선택하는 문제
4. 자재 운반 시스템이 대기 부품들 중에서 운반할 부품을 선택하는 문제

3.2.2. Priority function을 기초로 하는 FMS 통제 모델

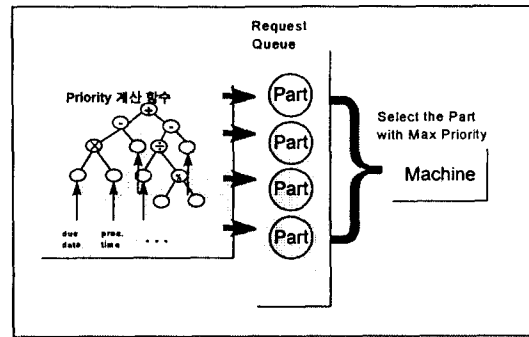
FMS 통제 모델은 다음과 같은 특징을 갖고 있다. 첫째, 기본적으로 시스템 내에 존재하는 각 entity들이 각각의 조건과 요구에 따라서 다른 entity에 request를 한다. 둘째, 특정의 entity에 이러한 request가 여럿일 경우 request queue를 형성한다. 셋째, 각 entity들이 이러한 request queue에서 특정 request를 선택하는 것들이 모여서 전체 시스템의 schedule을 형성한다. 넷째, 각 entity가 request queue에서 하나의 request를 선택하는 것은 각 request들에 대한 평가는 해당하는 priority function에 의하여 수행한다.

이러한 각각의 entity 사이의 해당 조건에 따른 수행 내용을 간략하게 도시화하면 [그림 4]와 같다.



[그림 4] FMS 통제 모델

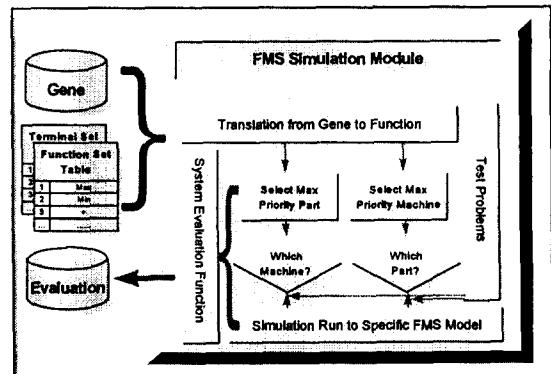
크게 네 가지의 의사결정 문제가 일어나게 된다. 즉, 1) 부품이 대안 기계들 중에서 다음 공정을 수행할 기계를 고르는 것, 2) 자신을 운반할 자재운반시스템을 고르는 것, 3) 기계가 다음 가공할 부품을 고르는 것, 4) 자재운반시스템이 다음 운반을 수행할 부품을 고르는 것 등은 각 대안들 중에서 해당 priority function의 값이 큰 것을 선택하게 된다. 이러한 과정들 중에서 기계에서 다음 가공할 부품을 선택하는 과정을 도시화 하여 [그림 5]에 예를 들어 나타내었다.



[그림 5] Priority Function의 사용 예 (기계에서 다음 가공 부품 선택)

3.2.3. External scheduling function embedded simulation

본 연구에서 제안하는 scheduling을 위한 priority function의 생성 방법론 상에서 각각의 priority function을 평가하는 데에는 적용 대상 FMS에 대한 simulation을 수행하는 것을 통하여 이루어진다. 이 때에 대상 FMS에 대한 기본적인 모델링의 동일 프레임 상에서 scheduling 방법으로 제시되는 새로운 priority function들이 대상 FMS의 전체 성능에 미치는 영향을 파악하여야 한다. 대상 시스템의 기본적인 모델링에는 수정을 하지 않고도 하위 수준의 운영 방식만의 변경은 주어진 scheduling 방법을 입력받아서 기본 모델에는 영향을 주지 않고 자동으로 simulation을 수행할 수 있는 방법론이 필요하다.

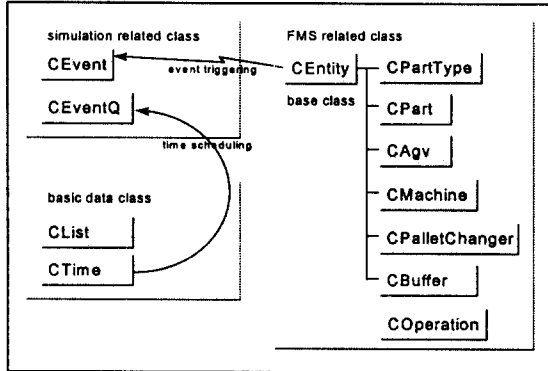


[그림 6] FMS simulation module

Simulation 모듈의 외부에서 입력받은 다양한 scheduling function을 수용할 수 있는 simulation 방법이다. (External Scheduling Function Embedded Simulation)

이러한 방법을 테스트하기 위하여 event-driven simulation을 수행하기 위한 simulation engine을 만들었고 FMS를 구성하는 각 entity들의 행동 방식을 Object-Oriented 개념을 활용하여 C++로 구현하였다. FMS simulation 모듈을 구성

하기 위해서 만든 클래스의 기본적인 구조는 다음의 그림과 같다.

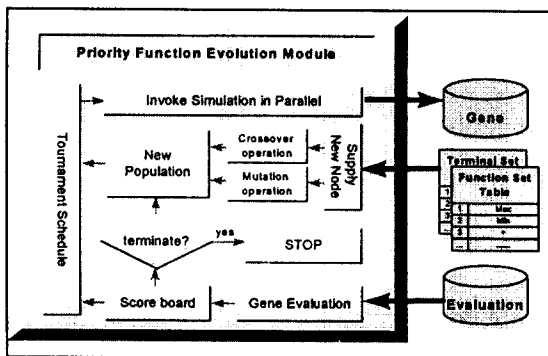


[그림 7] FMS simulation 모듈을 구성하는 클래스들

FMS와 관련된 entity들의 클래스를 정의하기 이전에 base class로 CEntity를 씌워서 다른 entity를 새로 정의하기 용이하도록 구성하였고, 또한 event-driven simulation을 수행하기 위한 일반화시킨 event-time관리용 클래스로 CEvent와 CEventQ를 정의하였다. 또한 공통적으로 자주 사용되는 자료 구조들에 대한 클래스로 CTime과 CList를 부수적으로 정의하였다.

3.3. Priority Function Evolution 모듈

본 연구에서 제시하는 전체 시스템의 두 개의 모듈 중의 하나인 Priority Function Evolution 모듈은 앞 장에서 제시한 FMS Simulation 모듈에서 사용될 priority function들을 탐색하고 생성하기 위한 모듈이다. 그러한 방법론으로는 앞에서 언급하였듯이 genetic programming을 사용하게 된다.



[그림 8] Priority Function Evolution 모듈

3.3.1. Gene Operator의 정의

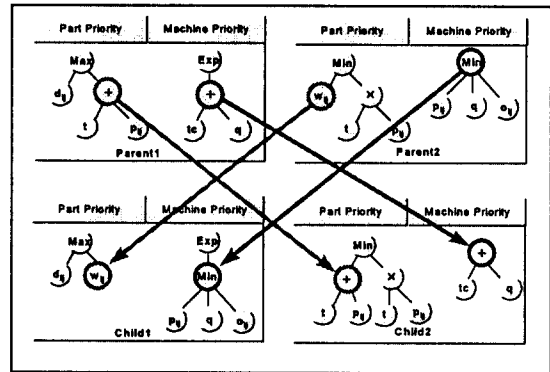
기본적으로 genetic programming과 달리 여러 entity들의 의사결정을 수행하기 위해서 필요한

여러 개의 priority function이 하나의 gene 벡터를 형성하게 된다. 즉, 아래의 수식과 같이 표현할 수 있다.

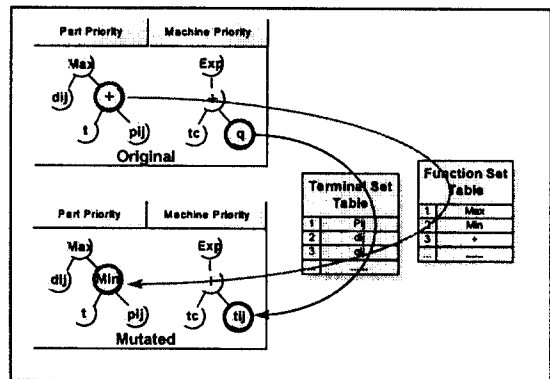
$$\text{Gene vector} = (\text{Gene_of_Part_Function}, \text{Gene_of_Machine_Function}, \dots)$$

따라서 일반적인 genetic programming에서의 gene operator를 그대로 적용시킬 수 없고 각 gene operator들을 gene 벡터의 각 차원들 끼리 수행시키는 방법을 적용하였다.

그에 따라서 아래의 [그림 9], [그림 10]에서 보여주듯이 각 부모에서 같은 차원들의 function들 끼리 operator를 적용시켜서 새로운 gene 벡터를 생성시키게 된다.



[그림 9] crossover operation



[그림 10] mutation operation

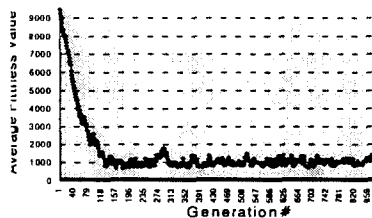
4. 시스템 적용

4.1. 시스템 적용

예제 FMS는 Horizontal machining center 3대, 5 axis machining center 1대, setup station 3대, washing machine 1대로 구성되어 있으며 AS/RS가 중앙 버퍼의 역할을 수행한다. 생산하는 부품은 총 20종류이고, Fitness function은 make-span값이며, 부품 종류별 생산량 투입 부하를 여섯 가지 중

류로 변화시켜 보았다. 연간 총생산량을 하루 생산량으로 또는 12시간 1 shift 작업량으로 가정한 것이며, 각 생산하는 부품의 비율을 변경시킨 자료들을 대상으로 실험을 실시하였다.

각 population들의 fitness value의 변화 추세를 [그림 9]와 같다. 전체 10000번의 generation을 실시하였고 초기에 빨리 수렴되는 특징을 보였다. 어느 정도 make-span이 감소된 이후에는 더 이상 감소되지는 않았다. 따라서 종료 기준으로 특정 방법보다는 실험을 통하여 make-span이 한 계치에 도달하면 종료시켰다.



[그림 9] generation에 따른 성능 추세

여섯 종류의 실험 집합에 대하여 본 연구 결과를 비교하기 위하여 일반적인 dispatching rule 중 다섯 가지와 비교하였다. Genetic programming을 위해 사용한 function set과 terminal set은 다음과 같다. primitive function의 선택은 기존의 변형된 dispatching rule에서 사용되는 기본적인 함수들만을 사용하였다.

$F = \{ \text{plus, minus, multiple, divide, min, max, sign} \}$

$$T_p = \left\{ p_i, p_{i+1}, i, w_i, \sum_{k=1}^m p_k, m-i, \sum_{k=1}^m p_k, m, \sum_{k=1}^i p_k, i-1, t, m, c \right\}$$

i: 현재 공정 인덱스 (1~m)

m: 전체 공정수

p_i : i 번째 공정 소요 시간

w_i : i 번째 공정을 위한 대기 시간

t: 시스템에서 지내온 시간

c: 상수

$$T_m = \{ d_i, b_i, qm_i, a_i, p_i, m_i, rq_i, c \}$$

i: 현재 공정 인덱스

d_i : idle 이면 1, 아니면 0

b_i : local buffer가 full이면 1, 아니면 0

qm_i : local buffer 용량

q_i : 현재 queue 크기

p_i : 가공 시간

m_i : i번 공정 기계까지의 이동시간

rq_i : request queue 크기

c: 상수

실험 결과 dispatching rule보다 전체적으로 더 짧은 기간에 생산을 완료할 수 있었고, dispatching 보다 평균 92%로 make-span이 감소되

었다. 실험 결과를 [표2]에 정리하였고 이 때 생성된 priority 함수들을 [표3]에 나타내었다.

[표 2] dispatching rule들과의 makespan 비교

실험 set	Genetic	FIFO	LIFO	SPT	LPT	SPT/TOT	makespan 비율
1	1005	1194	1175	1170	1166	1215	0.86
2	953	1237	1206	1270	1223	1233	0.79
3	814	690	668	651	634	633	0.96
4	419	481	479	481	432	419	0.87
5	582	581	598	606	575	549	1.06
6	1151	1194	1175	1170	1166	1215	0.98

[표 3] 생성된 Priority 함수들

Set	Part type	part qty	Best priority functions
1	20	75	Part gene : Divide(NestProcTime, MovingTime) Machine gene : Minus(CurCLength, CurCLength)
2	10	75	Part gene : Min(Divide(MovingTime, CurProcNum), Divide(NestProcTime, Min(Divide(MovingTime, Min(CurProcNum, Min(Divide(NestProcTime, NestProcTime), Min(PastProcNum, CurProcNum))))), Min(PastProcNum, CurProcNum))) Machine gene : 1
3	10	75	Part gene : Plus(CurWaitTime, PastLieTime) Machine gene : Divide(CurCLength, Minus(Processing, MaxCLength))
4	3	18	Part gene : Max(TotProcTime, NestProcTime) Machine gene : Minus(Minus(ReqCSize, CurCLength), ReqCSize)
5	20	25	Part gene : Max(PastProcNum, Multiple(Max(TotProcTime, PastProcNum), Multiple(CurProcTime, Max(TotProcTime, CurProcTime)))) Machine gene : Sign(IsBlocking, CurCLength)
6	20	75	Part gene : Multiple(PastProcTime, Multiple(PastProcTime, Multiple(Multiple(PastProcTime, PastProcTime), Multiple(PastProcTime, 1)))) Machine gene : Sign(IsProcessing, ReqCSize)

5. 결론

본 연구에서는 적용 시스템에 적합한 특성을 갖는 priority function을 기반으로 하는 FMS scheduling 방법을 자동으로 생성시키기 위한 시스템을 제안하고 그 가능성을 검증하였다. 전체의 시스템은 크게 Function Evolution 모듈과 FMS Simulation 모듈로 구성된다. Function Evolution 모듈에서는 genetic programming을 사용하여 새로운 priority function 들을 탐색해 내는 역할을 수행하고 FMS Simulation 모듈에서는 생성된 priority function 들을 평가하는 역할을 수행하게 된다.

본 연구에서 제안하는 시스템은 기존의 FMS scheduling에 대한 연구들이 해를 구하는 방법을 찾은 것이 중점을 둔 반면에 scheduling 방법을 찾은 체계적인 방법을 제시함으로써 다양한 시스템에 대해서 scheduling 방법을 생성시키는 데에 도움을 줄 수 있게 되었다.

6. 참고문헌

- [1] 강석호, "생산·운영관리", 경세원, 서울, (1994)

- [2] Akturk, M.S., Yilmaz, H., "Scheduling of Automated Guided Vehicles in a Decision Making Hierarchy", *IJPR*, Vol.34, No.2, pp.577-591, (1996)
- [3] Basnet, C., Mize, H.J., "Scheduling and Control of Flexible Manufacturing Systems: a Critical Review", *IJCIM*, Vol.7, No.6, pp.340-355, (1994)
- [4] Buzacott, J.A., Yao, D.D., "Flexible Manufacturing Systems: A Review of Analytical Models", *Management Science*, Vol.32, No.7, pp.890-905, (1986)
- [5] Chand, S., Traub, R., Uzsoy, R., "An Iterative Heuristic for the Singles Machine Dynamic Total Completion Time Scheduling Problem", *Computers and Ops. Res.*, Vol.23, No.7, pp.641-651, (1996)
- [6] Cho, H., Wysk, R.A., "A Robust Adaptive Scheduler for an Intelligent Workstation Controller", *IJPR*, Vol.31, No.4, pp.771-789, (1993)
- [7] Dorndorf, U., Pesch, E., "Evolution based learning in a job shop scheduling environment", *Computers and OR*, Vol.22, No.1, pp.25-40, (1995)
- [8] Egbelu, P. J., Tanchoco, J. M. A., "Characterization of Automated Vehicle Dispatching Rules", *IJPR*, Vol.22, No.3, pp.359-374, (1984)
- [9] Goyal, S.K., Mehta, K., Kodali, R., Deshmukh, S.G., "Simulation for Analysis of Scheduling Rules for a Flexible Manufacturing System", *IMS*, Vol.6, No.5, pp.21-26, (1995)
- [10] Ishii, N., Talvage, J. J., "A Mixed Dispatching Rule Approach in FMS Scheduling", *IJFMS*, Vol.6, pp.69-87, (1994)
- [11] Joohee, K., Funk, H. K., Fichter, F. E., "Towards an expert system for FMS scheduling: a knowledge acquisition environment", *Expert systems and intelligent manufacturing*, edited by michael D. Oliff, (1988)
- [12] Koza, J. R., "Genetic programming: on the programming of computers by means of natural selection", MIT Press, Cambridge, Massachusetts, (1992)
- [13] Kusiak, A., Chen, M., "Expert systems for planning and scheduling manufacturing systems", *EJOR*, Vol.34, pp.113-130, (1988)
- [14] Liu, J., MacCarthy, B. L., "The Classification of FMS Scheduling Problems", *IJPR*, Vol.34, No.3, pp.647-656, (1996)
- [15] Prakash, A., Chen, M., "A Simulation Study on Flexible Manufacturing Systems", *Computers on Industrial Engineering*, Vol.28, No.1, pp.191-199, (1995)
- [16] Ramaswamy, S.E., Joshi, S.B., "A Response Surface Approach to Developing Composite Dispatching Rules for Flexible Manufacturing Systems", *IJPR*, Vol.32, No.11, pp.2613-2629, (1994)
- [17] Rickel, J., "Issues in the design of scheduling systems, Expert systems and intelligent manufacturing", edited by michael D. Oliff, (1988)
- [18] Tempelmeier, H., Kuhn, H., "Flexible Manufacturing Systems : Decision Support for Design and Operation", John Wiley and Sons, New York, (1993)
- [19] Tincich, J. D., "CHESS: a methodology for resolving scheduling and dispatch problems in FMSs", *IJFMS*, Vol.8, 23-43, (1996)
- [20] Tzafestas, S., Triantafyllakis, A., "A New Adaptively Weighted Combinatorial Dispatching Rule for Complex Scheduling Problems", *CIMS*, Vol.7, No.1, pp.7-15, (1994)
- [21] Vepsalainen, A.P.J., Morton, T.E., "Priority Rules for Job Shops With Weighted Tardiness Costs", *Management Science*, Vol.33, No.8, pp.1035-1047, (1987)