

유전 알고리즘을 이용한 두 가지 목적을 가지는 스케줄링의 최적화[☆]

Optimization of Bi-criteria Scheduling using Genetic Algorithms

김 현 철*
Hyunchul Kim

요 약

멀티 프로세서 시스템에서 스케줄링은 매우 중요한 부분이지만, 최적의 해를 구하는 것이 복잡하여 다양한 휴리스틱 방법들에 의한 스케줄링 알고리즘들이 제안되고 있다.

최근 유전 알고리즘을 사용한 멀티 프로세서 스케줄링 알고리즘들이 제시되고 있지만, 제시된 알고리즘 대부분은 한가지만의 목적을 가지는 단순한 알고리즘이다.

본 논문에서는 유전 알고리즘을 이용한 새로운 스케줄링 알고리즘을 제시한다. 또한, 해를 구하는 과정에서 시뮬레이티드 어닐링 (simulated annealing)의 확률을 이용하여 유전 알고리즘의 성능을 개선시킨다. 제시된 알고리즘은 태스크들의 최종 수행 완료 시간 (makespan)을 최소화하는 것과 사용된 프로세서의 수를 최소화하는 두 가지의 목표를 가진다.

모의 실험을 통하여 제시된 알고리즘이 다른 알고리즘보다 최종 수행 완료 시간과 사용된 프로세서의 수에서 더 나은 결과를 보임을 확인할 수 있었다.

Abstract

The task scheduling in multiprocessor system is one of the key elements in the effective utilization of multiprocessor systems. The optimal assignment of tasks to multiprocessor is, in almost all practical cases, an NP hard problem. Consequently various modern heuristics based algorithms have been proposed for practical reason.

Recently, several approaches using Genetic Algorithm (GA) are proposed. However, these algorithms have only one objective such as minimizing cost and makespan.

This paper proposes a new task scheduling algorithm using Genetic Algorithm combined simulated annealing (GA+SA) on multiprocessor environment. In solution algorithms, the Genetic Algorithm (GA) and the simulated annealing (SA) are cooperatively used. In this method, the convergence of GA is improved by introducing the probability of SA as the criterion for acceptance of new trial solution. The objective of proposed scheduling algorithm is to minimize makespan and total number of processors used.

The effectiveness of the proposed algorithm is shown through simulation studies. In simulation studies, the results of proposed algorithm show better than that of other algorithms.

⇨ Keyword : genetic algorithm, multiprocessor scheduling, simulated annealing

1. 서 론

최근 멀티 프로세서 시스템에서의 응용프로그램이 급속도로 발전되고 있음에 따라 많은 태스

크들을 보다 빨리 처리하기 위해 스케줄링에 대한 관심이 높아지고 있으나 멀티 프로세서 시스템에서의 태스크 스케줄링은 매우 복잡하기 때문에 유전 알고리즘 (genetic algorithm GA)을 포함한 다양한 휴리스틱 (heuristic) 방법을 이용한 알고리즘들이 제시되고 있다[1].

유전 알고리즘은 생물의 적자 생존 과정을 이용한 최적화 수법으로 전통 수리적인 방법으로 해결하기 어려운 많은 문제들에 적용되어 좋은

* 정 회 원 : 경주대학교 컴퓨터 멀티미디어공학부 교수
kimhc@kyongju.ac.kr

[2005/04/01 투고 - 2005/06/27 심사 - 2005/09/21 심사완료]

☆ 본 연구는 경주대학교 2005년도 학술연구비 지원에 의한 것입니다.

결과들을 보여왔다. Mitra와 Ramanatha는 유전 알고리즘을 이용하여 태스크간의 실행순서와 마감시간을 가진 비선점 태스크들을 위한 스케줄링 알고리즘을 제시하였고[2]. Lin과 Yang은 조합 (Hybrid) 유전 알고리즘을 사용하여 실행 순서별로 정렬된 태스크들을 위한 스케줄링 알고리즘을 제시하였으며[3], Monnier는 유전 알고리즘을 사용하여 실시간 태스크를 위한 스케줄링 알고리즘을 제시하였다[4]. 또한, 프로세서간의 성능이 상이한 환경에서의 스케줄링에 대한 알고리즘도 제시되었다[5-7]. 그러나 이러한 알고리즘들은 최종 수행 완료시간을 최소화하거나 비용을 최소화하는 등 모두 한 가지만의 목적을 가지고 있다. 실제의 응용 프로그램에서는 동시에 다수의 목적을 만족시키는 스케줄링 알고리즘이 필요하므로 한 가지의 목적에 맞게 설계된 스케줄링 알고리즘은 실제로 적용되기에는 다소 부족한 점이 있다.

본 논문에서는 유전 알고리즘을 기본으로 하는 새로운 스케줄링 알고리즘을 제시한다. 해를 구하는 과정에서 시뮬레이티드 어닐링 (simulated annealing)[8]의 확률을 이용하여 유전 알고리즘의 성능을 개선시킨다. 제시된 알고리즘은 태스크들의 최종 수행 완료 시간 (makespan)을 최소화하는 것과 사용된 프로세서의 수를 최소화하는 두 가지의 목표를 가진다.

본 논문의 구성은 다음과 같다.

2절에서는 멀티 프로세서 스케줄링 문제를 보다 자세히 설명하고, 멀티 프로세서 스케줄링 문제를 수학 모델화한다. 3절에서는 유전 알고리즘과 시뮬레이티드 어닐링의 확률을 설명하고 제시된 알고리즘을 소개한다. 4절에서는 제안한 알고리즘의 성능을 측정하기 위해 수행한 모의 실험 결과를 보이고, 5절에서 결론을 맺는다.

2. 멀티 프로세서 스케줄링 문제와 수학 모델

본 논문에서 제시하는 스케줄링 알고리즘은

모든 태스크들의 최종 수행 완료 시간을 최소화하고 사용된 프로세서의 수를 최소화하는 두 가지의 목적을 가지고, 스케줄링 문제를 보다 명확히 정의하기 위해 다음과 같은 몇 가지 사항들을 가정한다.

1. 모든 태스크들은 비선점이다.
2. 모든 프로세서는 어떤 한 시점에서는 오로지 한 태스크만을 수행할 수 있다.
3. 모든 태스크는 동시에 두 개 이상의 프로세서에서 수행될 수 없다.
4. 프로세서들의 성능은 모두 같다.
5. 프로세서간의 메시지 전송시간은 무시한다.

이러한 멀티 프로세서 스케줄링 문제는 다음과 같이 수학적인 모델로 나타내어 질 수 있다. 모델에서 모든 태스크들의 수행 시간은 미리 알려져 있음을 가정한다.

$$\min f_1 = M \tag{1}$$

$$\min f_2 = \max\{t_i^S + c_i\} \tag{2}$$

$$\text{s.t. } t_i^S \geq t_j^S + c_j, \quad e_{ij} \in E \tag{3}$$

$$1 \leq M \leq N \tag{4}$$

위 수식에서 사용된 기호들을 다음과 같이 정의한다.

- 첨자

i, j : task index, $i, j=1,2,\dots,N$

m : processor index, $m=1,2,\dots,M$

- 파라메타

$G=(T, E)$: task graph

$T=\{t_1, t_2, \dots, t_N\}$: a set of N tasks

$E=\{e_{ij}\}, i, j=1,2,\dots,N, i \neq j$: a set of directed edges among the tasks representing precedence relationship

t_i : the i th task, $i=1,2,\dots, N$

e_{ij} : precedence relationship from task t_i to task t_j

p_m : the m th processor, $m=1,2,\dots,M$
 c_i : computation time of task t_i
 $pre^*(t_i)$: set of all predecessors of task t_i
 $suc^*(t_i)$: set of all successors of task t_i
 $pre(t_i)$: set of immediate predecessors of task t_i
 $suc(t_i)$: set of immediate successors of task t_i
 - 결정변수
 t_i^S : start time of task t_i
 M : total number of processors used

수식 (1)과 (2)는 제시하는 알고리즘의 목적 함수이다. 수식 (1)은 사용된 프로세서의 수 f_1 을 최소화하는 것을 의미하고, 수식 (2)는 모든 태스크들의 최종 수행 완료 시간 f_2 를 최소화하는 것을 의미한다. 수식 (3), (4)는 제시된 수학적 모델의 제약 조건들이다.

3. 시뮬레이티드 어닐링과 결합된 유전 알고리즘을 이용한 스케줄링

본 논문에서 제시하는 스케줄링 알고리즘은 유전 알고리즘을 기본으로 하고 있다. 유전 알고리즘은 생물의 적자 생존 과정을 이용한 최적화 수법이다 [9]. 유전 알고리즘의 표현 (encoding) 과 복원 (decoding) 등 멀티 프로세서 스케줄링에 사용된 알고리즘과 유전 알고리즘과 결합된 시뮬레이티드 어닐링 방법을 이 절에서 설명한다.

3.1 표현과 복원

염색체 (chromosome) V_k , $k=1,2,\dots,pop\ size$ 는 프로세서에 할당된 태스크들을 나타낸다. 여기에서 $pop\ size$ 는 한 세대에서의 전체 염색체의 수를 의미한다. 염색체 V_k 는 $u(\cdot)$ 와 $v(\cdot)$ 두 부분으로 나뉘어진다. $u(\cdot)$ 는 태스크들의 수행순서를 나타내고, $v(\cdot)$ 는 태스크들의 프로세서 할당 정보를 나타낸다. 염색체 각 부분의 길이는 태스크의 총 수와 같다.

표현 절차는 $u(\cdot)$ 부분의 생성을 위한 단계 1 과 $v(\cdot)$ 부분의 생성을 위한 단계 2의 두 단계로 이루어진다. 각 단계별 절차는 다음과 같다.

procedure : Encoding Strategy I for multi-processor scheduling **begin**

```

 $l \leftarrow 1, W \leftarrow \emptyset$  ;
while ( $T \neq \emptyset$ )
     $W \leftarrow W \cup \arg\{ \tau_i \mid pre^*(\tau_i) = \emptyset, \forall i \}$  ;
     $T \leftarrow T - \{ \tau_i \}, i \in W$  ;
while ( $W \neq \emptyset$ )
     $j \leftarrow \text{random}(W)$  ;
     $u(l) \leftarrow j$  ;
     $W \leftarrow W - \{ j \}$  ;
     $pre^*(\tau_i) \leftarrow pre^*(\tau_i) - \{ \tau_i \}, \forall i$  ;
     $l \leftarrow l + 1$  ;
end
end
output  $u(\cdot)$  ;
end
    
```

procedure: Encoding Strategy II for multi-processor scheduling **begin**

```

 $l \leftarrow 1, M \leftarrow \begin{cases} 1, & \text{if } k = 1 \\ M_{k-1} + 1, & \text{if } 1 < k \leq pop - size; \end{cases}$ 
while ( $l \leq N$ )
     $m \leftarrow \text{random}[1:M]$  ;
     $v(l) \leftarrow m$  ;
     $l \leftarrow l + 1$  ;
end
output  $v(\cdot)$  ;
end
    
```

표현 절차 단계 1에서 W 는 선행 태스크들을 가지지 않는 태스크의 집합이다. 제시된 표현 절차 단계 1은 항상 태스크들의 주어진 수행 순서를 만족하여 적법한 염색체를 생성한다. 단계 2에서 사용된 프로세서의 총 수 M 은 염색체에 따라 그 값이 다르다. 초기에 M 의 값은 1이지

만 그 다음 염색체를 생성할 때는 값이 1씩 증가한다.

생성된 염색체는 복원 단계에서 해독되어 스케줄링 결과를 생성한다. 복원 단계에서는 최종 스케줄링 결과를 생성할 뿐만 아니라, 사용된 프로세서의 총 수와 태스크들의 수행 시작 시간 및 종료 시간을 계산하여 두 가지의 목적 함수 값을 계산한다.

3.2 평가함수와 선택

유전 알고리즘에서 염색체는 평가함수에 의해서 평가되어진다. 좋은 평가를 받은 염색체는 그렇지 못한 염색체에 비해 다음 세대에 선택될 보다 높은 기회를 가지게 된다.

본 논문에서 제시하는 스케줄링 알고리즘은 목적 함수가 두 가지이다. 두 가지의 목적을 가지는 문제의 최적 해를 구하기 위한 연구는 1960년부터 꾸준히 이루어져 왔으며, 최근에는 유전 알고리즘에서도 여러 목적을 가지는 문제의 해결을 위한 연구가 이루어지고 있다. 본 논문에서는 가중치 부여 (adaptive weight)의 방법에 의하여 두 가지 목적을 가지는 문제를 위한 염색체를 평가하도록 한다.

가중치 부여 방법에 의한 평가 함수를 위해 각 세대에서의 염색체 값들의 최대점 f^+ 과 최저점 f^- 을 다음과 같이 정의한다.

$$f^+ = \{f_1^{\max}, f_2^{\max}\} \quad (5)$$

$$f^- = \{f_1^{\min}, f_2^{\min}\} \quad (6)$$

여기에서 f_q^{\max} 와 f_q^{\min} 는 각각 q 번째 목적의 최대 값과 최소 값이며 두 가지의 목적을 가지는 본 논문에서는 다음과 같이 정의한다.

$$f_q^{\max} = \max_k \{f_q(V_k)\}, q=1,2 \quad (7)$$

$$f_q^{\min} = \min_k \{f_q(V_k)\}, q=1,2 \quad (8)$$

염색체 V_k 의 두 가지의 목적 함수 값에 가중치를 부여하여 결합한 목적 함수의 값은 다음과 같다.

$$F(V_k) = \sum_{q=1}^2 w_q f_q(V_k) = \sum_{q=1}^2 \frac{f_q(V_k)}{f_q^{\max} - f_q^{\min}} \quad (9)$$

여기에서 w_q 는 다음과 같이 정의된다.

$$w_q = \frac{1}{f_q^{\max} - f_q^{\min}}, q=1,2 \quad (10)$$

염색체 V_k 의 평가 함수는 다음과 같다.

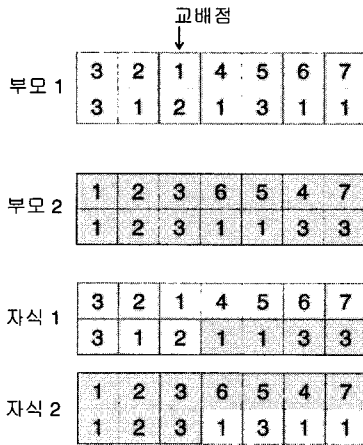
$$eval(V_k) = 1/F = \frac{1}{\sum_{q=1}^2 \frac{f_q}{f_q^{\max} - f_q^{\min}}} \quad (11)$$

선택 과정에서는 앞서 언급했듯이 룰렛 휠 (roulette wheel) [10] 방법을 사용한다.

3.3 교배와 변이

교배 (crossover)를 위하여 본 논문에서는 한 점 교배방법 (one-cut point crossover)을 이용한다. 태스크들의 수행 순서가 교배에 의하여 바뀌면 스케줄링의 조건으로 주어진 수행 순서를 위배하는 적합하지 않은 염색체가 생성될 위험이 있으므로 염색체의 두 부분 중 프로세서의 할당을 나타내는 $v(\cdot)$ 부분만을 교배하도록 한다. 그림 1은 사용된 교배 방법의 예를 보여준다.

염색체의 변이를 위해서는 가장 광범위하게



(그림 1) 염색체의 교배

사용되고 있는 한 점 변이 방법 (one-bit altering mutation)을 사용하였다.

3.4 시뮬레이티드 어닐링의 확률을 이용한 유전 알고리즘의 성능 개선

유전 알고리즘에 있어서 해의 수렴속도는 시뮬레이티드 어닐링의 확률에 의해 개선되어 질 수 있다. 시뮬레이티드 어닐링의 확률에 의한 유전 알고리즘의 개선에 대한 절차는 다음과 같이 나타내어 질 수 있다.

procedure: Improving of GA by the probability of SA **begin**

$r \leftarrow \text{random}[0,1];$

$\Delta E \leftarrow \text{eval}(V') - \text{eval}(V);$

if ($\Delta E > 0 \parallel r < \text{Exp}(\Delta E / Tem)$)

$V'' \leftarrow V;$

else

$V'' \leftarrow V;$

$Tem \leftarrow Tem \times \rho;$

output offspring chromosomes V''

end

제시된 절차에서 V 와 V' 는 각각 부모 염색체와 자식 염색체를 나타낸다. V'' 는 제시된 절

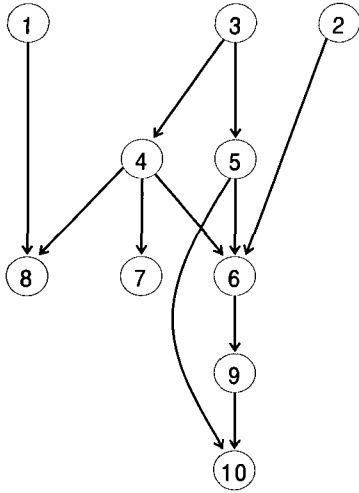
차에 의해 최종적으로 선택될 염색체를 의미한다. Tem 와 ρ 는 시뮬레이티드 어닐링에서의 온도와 냉각률을 나타낸다.

4. 모의 실험

본 논문에서 제시된 알고리즘의 성능을 검증하기 위해 모의 실험을 하였다. 모의 실험을 통해 제시된 알고리즘 (GA+SA)을 Monnier가 제시한 알고리즘 (Monnier's GA)과 시뮬레이티드 어닐링과 결합하지 않은 단순 유전 알고리즘 (simple GA)과 성능을 비교하였다. Monnier's GA는 단순한 유전 알고리즘에 기본을 두고 있으며, 선택에 있어서 선형 정규화 방법을 사용하고 있다. 선형 정규화 방법은 염색체들이 유사한 경우에 효과적으로 이용될 수 있으나 이러한 경우는 매우 제한적이다. 또한 Monnier's GA는 한 가지 목적을 위해 설계된 스케줄링 알고리즘이므로 본 논문에서 다루는 두 가지의 목적 함수 값을 동시에 비교하기에는 다소 무리가 있다. 각각의 목적 함수 값을 따로따로 평가하여 비교하도록 한다.

모의 실험에 사용된 태스크들은 랜덤하게 발생하였다. 태스크들의 수행 순서를 정하기 위하여 P-Method [11]를 사용하였다. P-Method는 인접행렬의 확률적 구성을 기본으로 한다. 태스크 t_i 에서 태스크 t_j 로의 수행 순서관계가 존재할 경우 행렬의 요소 a_{ij} 는 1이 된다. 그렇지 않고 독립적일 경우 행렬의 요소 a_{ij} 는 0이 된다. 행렬의 하삼각 요소들의 값은 항상 0이 된다. 행렬의 상삼각 요소들은 파라메타 ϵ 를 가지는 Bernoulli의 성공 확률 법칙을 바탕으로 하여 1과 0중 랜덤한 값을 가지게 된다. 파라메타 ϵ 가 0일 경우 모든 태스크들은 수행 선행관계를 가지지 않고, 파라메타 ϵ 가 1일 경우는 모든 태스크들이 그 수행 선행관계에 의해 서로 연결되어 있다.

태스크들의 수행 시간은 평균값 5의 지수분포에 의해 랜덤하게 발생시켰다.



〈그림 2〉 태스크 총 수 10의 수행 행관계

모의 실험은 총 태스크의 수 10, 50, 100의 세 가지 경우에 의해 실시되었다.

4.1 태스크의 총 수 10의 경우

모의 실험에 사용된 태스크들의 수행 선행관계는 그림 2와 같다.

그림 2에서의 태스크들의 수행시간은 다음의 표 1과 같다.

그림 3은 실험의 결과를 보여준다. 그림 3에서 본 논문에서 제시하는 스케줄링 알고리즘 GA+SA가 다른 두 개의 알고리즘보다 Pareto

〈표 1〉 태스크 총 수 10의 경우 태스크들의 수행 시간

i	suc(τ _i)	c _i	i	suc(τ _i)	c _i
1	8	5	6	9	2
2	6	6	7	-	3
3	4,5	11	8	-	4
4	6,7,8	10	9	10	6
5	6,10	10	10	-	11

해의 곡선이 이상점 (ideal point)에 더 가까운 것을 알 수 있다. 이 실험의 사용된 태스크들의 경우에 이상점은 $f_1=1, f_2=11$ 이다.

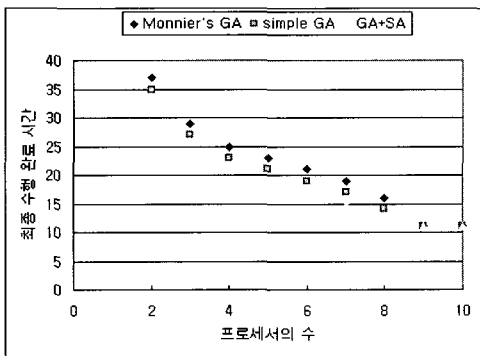
4.2 태스크의 총 수 50의 경우

모의 실험에 사용된 태스크들의 수행 선행관계를 나타내는 그래프 그림과 수행시간을 나타내는 표는 생략하도록 한다.

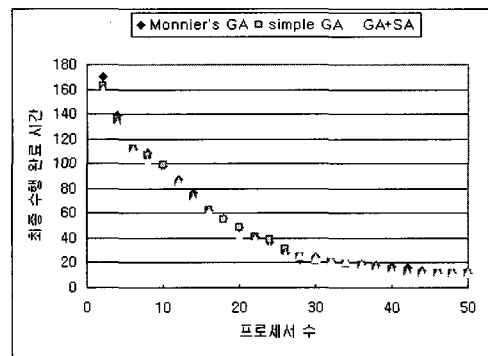
그림 4는 실험의 결과를 보여준다. 그림 4에서 또한 본 논문에서 제시하는 스케줄링 알고리즘 GA+SA가 다른 두 개의 알고리즘보다 Pareto 해의 곡선이 이상점 (ideal point)에 더 가까운 것을 알 수 있다. 이 실험의 사용된 태스크들의 경우에 이상점은 $f_1=1, f_2=12$ 이다.

4.3 태스크의 총 수 100의 경우

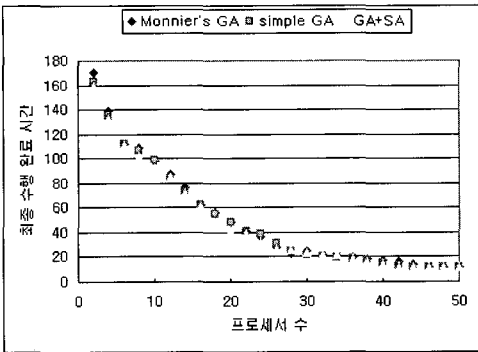
모의 실험에 사용된 태스크들의 수행 선행관



〈그림 3〉 태스크 총 수 10의 모의 실험 결과



〈그림 4〉 태스크 총 수 50의 모의 실험 결과



〈그림 5〉 태스크 총 수 100의 모의 실험 결과

계를 나타내는 그래프 그림과 수행시간을 나타내는 표는 생략하도록 한다.

그림 5는 실험의 결과를 보여준다. 그림 5에서 또한 본 논문에서 제시하는 스케줄링 알고리즘 GA+SA가 다른 두 개의 알고리즘보다 Pareto 해의 곡선이 이상점 (ideal point)에 더 가까운 것을 알 수 있다. 이 실험의 사용된 태스크들의 경우에 이상점은 $f_1=1, f_2=12$ 이다.

5. 결 론

멀티 프로세서 시스템에서의 태스크 스케줄링은 매우 복잡하기 때문에 다양한 휴리스틱 방법을 이용한 알고리즘들이 제시되고 있다.

최근, 유전 알고리즘 (genetic algorithm; GA)을 이용한 몇몇의 스케줄링 알고리즘들이 제시되었으나 이러한 알고리즘들은 최종 수행 완료 시간을 최소화하거나 비용을 최소화하는 등 모두 한 가지만의 목적을 가지고 있다. 한 가지의 목적에 맞게 설계된 스케줄링 알고리즘은 실제로 적용되기에는 다소 부족한 점이 있다. 실제의 응용 프로그램에서는 동시에 다수의 목적을 만족시키는 스케줄링 알고리즘이 필요하다.

본 논문에서는 유전 알고리즘을 기본으로 하는 새로운 스케줄링 알고리즘을 제시한다. 해를 구하는 과정에서 시뮬레이티드 어닐링 (simulated annealing) [8]의 확률을 이용하여 유전 알고리

즘의 성능을 개선시킨다. 제시된 알고리즘은 태스크들의 최종 수행 완료 시간 (makespan)을 최소화하는 것과 사용된 프로세서의 수를 최소화하는 두 가지의 목표를 가진다.

제시된 알고리즘은 모의 실험을 통하여 유전 알고리즘을 이용한 다른 스케줄링 알고리즘과 성능을 비교하였다. 모의 실험 결과에서 다른 스케줄링 알고리즘보다 제시된 스케줄링 알고리즘이 사용된 프로세서의 수와 최종 수행완료 시간이 현저히 작음을 확인할 수 있었다.

제시된 스케줄링 알고리즘은 기타 유사한 환경에서의 스케줄링 알고리즘, 예를 들어, 프로세서 간의 메시지 전송시간을 고려한 환경에서의 스케줄링 알고리즘 등으로 확장될 수 있을 것이다.

참고문헌

- [1] Yalaoui, F., and C. Chu, "Parallel Machine Scheduling to Minimize Total Tardiness", International Journal of Production Economics, vol. 76, no. 3, pp. 265-279, 2002.
- [2] Mitra, H. and P. Ramanathan, "A Genetic Approach for Scheduling Non-preemptive Tasks with Precedence and Deadline Constraints", Proc. of the 26th Hawaii International Conference on System Sciences, pp. 556-564, 1993.
- [3] Lin, M. and L. Yang, "Hybrid Genetic Algorithms for Scheduling Partially Ordered Tasks in A Multi-processor Environment", Proc. of the 6th International Conference on Real-Time Computer Systems and Applications, pp. 382-387, 1999.
- [4] Monnier, Y., J. P. Beauvais and A. M. Deplanche, "A Genetic Algorithm for Scheduling Tasks in a Real-Time Distributed System", Proc. of 24th

- Euromicro Conference, pp. 708-714, 1998.
- [5] Theys, M. D., T. D. Braun, H. J. Siegal, A. A. Maciejewski and Y. K. Kwok. Mapping Tasks onto Distributed Heterogeneous Computing Systems Using a Genetic Algorithm Approach. In Zomaya, A. Y., F. Ercal and S. Olariu, editors, Solutions to Parallel and Distributed Computing Problems, chapter 6, 135-178, John Wiley & Sons 2001.
- [6] Page A. J. and T. J. Naughton. Dynamic Task Scheduling using Genetic Algorithm for Heterogeneous Distributed Computing. In: Proceedings of 19th IEEE International Parallel and Distributed Processing Symposium 2005: 189.1.
- [7] Dhodhi, M. K., I. Ahmad and A. Yatama and I. Ahmad. An Integrated Technique for Task Matching and Scheduling onto Distributed Heterogeneous Computing Systems. Journal of Parallel and Distributed Computing 2002; 62: 1338-1361.
- [8] Kirkpatrick, S., C. D. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing", Science, vol. 220, no. 4598, pp. 671-680, 1983.
- [9] Kim, H. C, Y. Hayashi and K. Nara, "An Algorithm for Thermal Unit Maintenance Scheduling through combined use of GA, SA and TS", IEEE Transactions on Power Systems, vol. 12, no. 1, pp. 329-335, 1997.
- [10] Hou E. S. H., R. Hong and N, Ansari, "Efficient multiprocessor scheduling based on genetic algorithms", IEEE Conference on Industrial Electronics Society, vol. 2, pp.1239-1243, 1990.
- [11] Al-Sharaeh, S. and B. E. Wells, "A Comparison of Heuristics for List Schedules using The Box-method and Pmethod for Random Digraph Generation", Proc. of the 28th Southeastern Symposium on System Theory, pp. 467-471, 1996.

● 저자 소개 ●



김 현 철 (Kim, Hyunchul)

1982년 숭실대학교 전자계산학과 졸업(공학사)

1993년 숭실대학교 정보과학대학원 정보산업학과 졸업(이학석사)

1996년 이바라키대학교 대학원 공학연구과 졸업(공학박사)

2000년~현재 경주대학교 컴퓨터멀티미디어학부 교수

관심분야 : 모던휴리스틱스, 최적화 기법 등 Modern Heuristic, Optimization Technique etc.

E-mail : kimhc@gju.ac.krr