



뒤로 이동하는 경우

- 방법 2 : 자유공간 중 조각파일의 크기에 가장 적합한 곳으로 이동
- 방법 3 : 자유공간 중 가장 넓은 곳으로 이동
- 방법 4 : 자유공간 중 처음으로 조각파일 전체 크기에 맞는 곳으로 이동
- 방법 5 : 첫 조각을 중심으로 나머지 조각이 앞으로 이동하는 경우

2.2 교차 및 돌연변이

기본적인 유전연산자로 교차(crossover)와 돌연변이(mutation)연산을 사용한다.

1) 교차(crossover)

교차 연산자는 부분사상교차(PMX : Partially Mapped crossover)와 순환교차(CX : Cycle crossover)를 사용하였다[4].

2) 돌연변이(mutation)

돌연변이 연산자는 각 유전자에 대하여 두 개의 난수를 생성하여, 그 난수의 위치에 해당하는 유전자를 교환하는 방식의 돌연변이를 사용하였으며, 이동방법에 대한 돌연변이는 고려하지 않았다.

2.3 적합도 함수 및 선택

적합도 함수는 각각의 개체에 대하여, 각 유전자좌에 나타난 이동방법에 따라 조각파일을 이동하기 위한 원시 적합도를 계산하고 이들의 총합으로 한 개체의 적합도를 정한다. 즉, 검색체 상에 나타난 순서 및 이동방법에 따라 각 조각파일의 이동거리를 계산한 후 그 총합을 개체의 적합도로 정한다.

원시 적합도는 트랙(track)을 옮기는 시간과 섹터(sector)를 옮기는 이동시간(move time)으로 정한다. 일반적으로 트랙을 찾는 시간(seek time)이 섹터를 찾는 시간(latency time)보다 10배정도 더 길다는 것에 착안하여 상대적인 시간을 계산하였다[6].

선택은 가장 일반적인 적합도 비례선택을 사용하였으며, 이 문제의 경우는 이동거리가 짧을수록 좋은 것이기 때문에, 계산된 적합도의 역수를 취하여 비례선택을 하였다[1].

3. 성능 평가 및 분석

여기서는 조각파일이 디스크 상에 어떻게 분포되어 있는가에 따라 디스크의 상태를 구분하고, GA 적용시 사용되는 여러 가지 변수를 달리해가며, 순서를 고려하지 않는 조각모음과 GA를 적용한 조각모음을 비교분석 하였다. 또한, GA의 적용의 절대시간이 모집단의 크기, 최대 세대수, 및 조각파일의 개수에 따라 어떻게 변화하는가를 분석하여 보았다.

GA를 적용함에 있어서 이동방법을 2.1절의 5 가지를 모두 사용하는 경우와 보다 효율적으로 판단되는 방법 1,2 만을 사용하는 것으로 구분하였으며 각각의 경우도 이동시간이 최소인 경우로 선택하는 방법과 가능한 것 중에서 임의로 선택하는 것으로 나누어 실험하였다. 즉

이동방법을 정하는 원칙에 따라 다음과 같이 4 가지로 구분하여 실험하였다.

- 1) 이동방법 1, 2 에서 가능한 경우 중 작은 이동 시간을 해당 유전자의 적합도로 계산(최소2).
- 2) 이동방법 1, 2 에서 임의적(random)으로 선택하여 가능한 경우 중 작은 이동시간을 해당 유전자의 적합도로 계산(임의2).
- 3) 이동방법 5 가지에서 가능한 경우 중 작은 이동 시간을 해당 유전자의 적합도로 계산(최소5).
- 4) 이동방법 5 가지에서 임의적(random)으로 선택하여 가능한 경우 중 작은 이동시간을 해당 유전자의 적합도로 계산(임의5).

3.1 디스크 상태에 따른 성능평가

조각파일의 분포에 따른 디스크 상태에 따라 조각파일의 수, 모집단의 크기(population), 교차(crossover)확률, 돌연변이(mutation)확률별로 실험하였다. 평가요소는 모집단의 평균 이동시간과 최적개체의 이동시간(최소 이동시간)을 기준으로 순서를 고려하지 않는 방법과 비교 분석하여 보았다.

자료를 분석하기 위한 그래프는 모집단의 크기별로 하였으며, 교차와 돌연변이 확률에 따른 변화는 CASE1(교차 0.6, 돌연변이 0.01), CASE2(교차 0.6, 돌연변이 0.05), CASE3(교차 0.8, 돌연변이 0.01), CASE4(교차 0.8, 돌연변이 0.05)등으로 표기하였다.

파일이 디스크에 전체적으로 분포되어 있는 경우, 각각의 이동방법에 따라 조각파일수, 모집단의 크기, 교차 확률, 돌연변이확률을 달리하여 실험한 결과는 <그림 2,3,4>와 같다(그림에서 기준은 순서를 고려하지 않는 조각모음을 나타낸다).

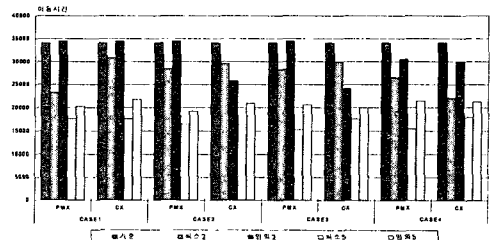


그림 2. 모집단의 크기 30의 최소이동시간

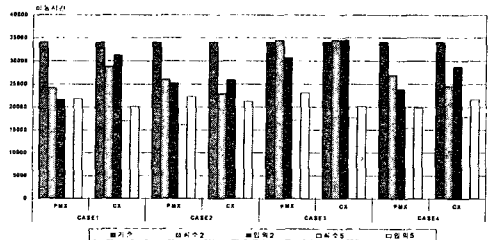


그림 3. 모집단의크기 50의 최소이동시간

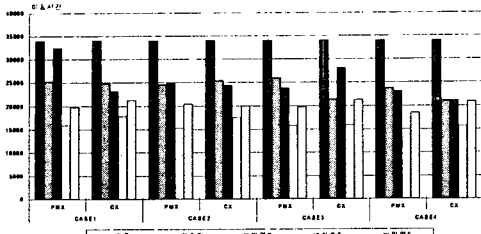


그림 4. 모집단의 크기 100의 최소이동시간

위의 결과와 같이 순서를 고려하지 않는 조각파일의 이동시간과 GA를 적용한 조각파일 순서에 따른 이동시간 중 '최소5' 방식이 평균적으로 50%이상의 성능향상을 보이고, '최소2', '임의2', '최소5'는 대부분의 경우 10 ~ 30% 정도의 성능향상을 보임을 알 수 있다. 결국 '최소5'를 적용하여 최적개체에 의해 조각모음을 하면 50% 이상 성능이 향상됨을 알 수 있다.

3.2 이동시간에 따른 성능평가

여기서는 GA 적용의 절대시간이 모집단의 크기, 최대 세대수, 및 조각파일의 개수에 따라 어떻게 변하는가를 평가한다. 실험에서는 동일한 조건하에서 10회의 실험을 반복하여 평균 수행시간(평균시간)과 최적 개체를 찾을 때까지의 소요시간(최소시간)을 계산하였다.

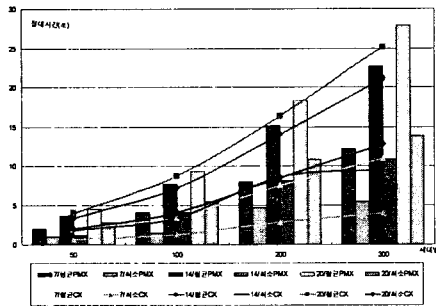


그림 5. 세대수 증가에 따른 PMX와 CX의 GA수행시간 비교

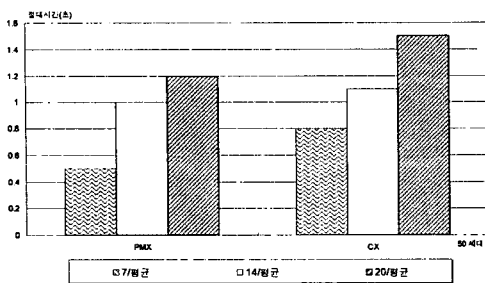


그림 6. 파일수 증가에 따른 PMX와 CX의 GA 수행시간 비교

<그림 5, 6>은 GA의 적용시 조각파일수 및 세대수가 증가됨에 따른 GA 수행시간이 기울기 1 이하의 완만한 직선을 따른다는 것을 나타낸다. 따라서, 조각파일 수가 늘어나고 세대수가 증가되어도 GA 수행시간이 지수적으로 증가되지 않을 것으로 판단된다.

4. 결론

제안한 방법에 대한 실험결과 순서를 고려하지 않는 조각모음 방법에서 조각파일의 이동시간에 비하여 GA를 적용한 조각모음에서의 조각파일 이동시간이 평균적으로 50%이상 줄어든다는 것을 알 수 있었다. 또한, GA 적용의 절대시간이 모집단의 크기, 최대 세대수, 및 조각파일의 개수에 따라 어떻게 변하는가를 평가하였다. 실험 결과 GA의 적용시 조각파일수 및 세대수가 증가됨에 따른 GA 수행시간이 기울기 1 이하의 완만한 직선을 따른다는 것을 알 수 있었다. 따라서, 조각파일수가 늘어나고 세대수가 증가되어도 GA 수행시간이 지수적으로 증가되지 않을 것으로 판단된다.

후추 연구과제는 디스크 전체 조각모음을 위한 GA 운용 및 시스템을 관찰하여 자주 사용되는 파일은 특별한 곳에 모으고, 관련된 파일들의 이웃된 곳으로 모으는 조각모음에 GA를 적용하는 것이다.

5. 참고문헌

- [1]. David B. Fogel, Lawrence J. Fogel, " *An Introduction to Evolutionary Programming*", Natural Selection Inc.
- [2]. Goldberg, D. E., " *Genetic Algorithms in Search, Optimization, and Machine Learning* ", Addison-Wesley Publishing Company Inc., pp.1-22,1989.
- [3]. Goldberg, D.E. and Ligte, RJr., " *Alleles, Loci and the Traveling Salesman Problem* ", Proc.1st. ICGA, 1985.
- [4]. Jose L.Ribeiro Filho, Philip C.Treleaven, Cesare Alippi, " *Genetic - Algorithm Programming Environments* ", IEEE Computer, pp.28-43, June 1994.
- [5]. Melanie Mitchell, Stephanie Forrest, " *Genetic Algorithms and Artificial Life*", The MIT Press Cambridge, Massachusetts London, England, pp.267-289, 1995.
- [6]. Peter Avel, " *IBM PC ASSEMBLY LANGUAGE AND PROGRAMMING* ", Prentice-Hall, Inc, pp.341-355, 1998.
- [7]. Zbigniew Michalewicz, " *Genetic Algorithms + Data Structures = Evolution Programs*", Springer-Verlag, pp.209-237, 1996.
- [8]. 강태원, " 혼돈 동역학계를 이용한 혼돈 진화 탐색 전략과 수정 오류 역전파 알고리즘에의 응용 ", 박사논문, 고려대학교, 1996.