

유전자 알고리즘을 이용한 물류에서의 지능적 운송 자원 할당

Intelligent Allocation of Transporting Resources in Logistics using Genetic Algorithm

김주원, 차영필, 정무영

포항공과대학교 산업공학과 / 제품생산기술연구소

Abstract

Recently, most of countries in the world are investing huge amount of capital for the infrastructure of logistics and trying to gain dominating position in logistics. To play the role of important hub in logistics, an efficient, flexible, and fault-tolerant transportation process should be developed. Minimization of transportation cost and timely deliveries in the unpredictable environment are a few of the important issues in logistics. This study suggests a way of transporting goods to destinations at the minimal cost and with the minimal delay by optimally allocating transporting resources. Various attributes in transportation such as due date, priority etc. are also considered. Appropriate transporting resources for each work item is selected by calculating the weighted sum of the cost factor and the delay factor assuming that initial sequences of work items are given. A policy to reallocate transporting resources is also suggested when work items or transporting resources are added or deleted because of accidents or disturbances. This policy provides adaptability to the allocation methodology which enables the system to cope with changing environment by controlling various attributes in transportation. Genetic algorithm is used for this approach.

Keywords : logistics, transporting resources, resource allocation, genetic algorithm

1. Introduction

최근 세계 여러 나라들이 물류 중심국의 위치를 차지하기 위하여 많은 노력을 기울이고 있다. 물류 중심지로서의 역할을 하기 위해서는 각 지역으로의 운송을 효율적으로 차질 없이 진행시켜야 한다. 즉, 운송비의 최소화와 운송납기일의 준수가 중요한 논점이 된다. 우리나라의 국가 물류비는 GDP의 12.8%(66.7조원, 2000년 기준)로 선진국 수준(GDP의 10%)보다 상당히 높은 실정이다. 이에 정부는 2010년까지 물류비를 선진국 수준인 10%로 절감하기 위하여 물류 시설의 확충과 물류 체계의 효율화에 많은 노력을 기울이고 있다 [1].

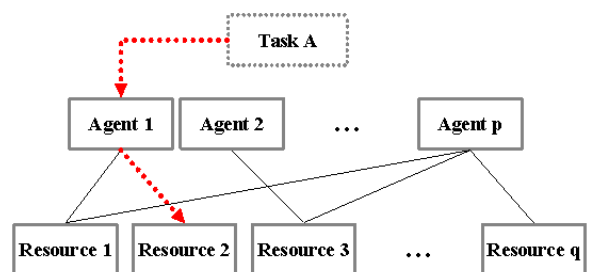
현재 물류 체계의 효율화에 관련된 연구들이 많이 진행되고 있는데, 그 중 Baita(2000)[2]는 제한 조건을 만족 시키면서 목적식을 최대화 만들어 주는 운송수단을 정해진 일정의 여정에 할당하는 문제에 관하여 연구하였다. 그는 버스 서비스의 구체적인 실제적인 상황에 대하여, Mathematical Approach, Heuristic Technique, AI Technique 등 다양한 방법을 통해 문제에 접근했고 효율성을 비교하였다.

또한 Task의 Scheduling과 Allocation에 관련된 문제를 Genetic Algorithm을 포함한 다양한 AI Technique을 이용해 접근한 연구가 많이 진행되었다. Genetic Algorithm은 1970년대 중반 Holland에 의해서 만들어진 최적화 기법의 일종으로서, 최근에는 Machine Learning, 최단경로 찾기, 자원 할당 등의 여러 가지 최적화 문제에 폭넓게 적용되고 있다. Lee(2003)[3]는 Multi-Agent System에 있어서 Agent들이 서로 조정하여 Task들을 할당해 나가는 과정을 Genetic Algorithm으로 구현하였다. 모든 Task들은 Subtask들의 합으로 구성되며, 이러한 Subtask들을 각 Agent에게 할당하여 최대의 Performance를 낼 수 있는 알고리즘을 개발하였다. 그는 또한 Genetic Algorithm을 적용하는 과정에서 Upper-Layer와 Lower-Layer의 Two-Layered Genetic Algorithm을 사용하여, Task Allocation과 Task Scheduling을 수행하였다. David(1999)[4]는 Cost의 최소화, Utilization의 최대화 등을 포함하는 여러 가지 목적을 달성하는 Scheduling에 Genetic Algorithm을 적용하였다.

본 연구에서는 물류에 있어서, 납기일을 최대한 지키면서 최소의 비용으로 운송이 가능한 방안을 각 작업과 운송 자원의 특성을 고려하여 모색해 보고자 한다. 또한 작업이나 운송 자원이 추가 또는 제거되었을 경우에, 새롭게 운송 자원을 할당해 주는 정책에 대해서도 고려해 본다. 뿐만 아니라 가장 적절한 작업 순서에 대해서도 고려해 본다.

2. Problem Description

운송 자원을 공유하는 여러 물류 회사가 있을 때, 각 물류 회사는 주어진 작업들에 대하여 필요한 운송 자원을 할당해야 한다. [그림 1]은 각 물류 회사의 작업에 자원을 할당하는 과정을 나타낸 것이다. 여기서 Agent는 각각의 물류회사를 나타내고 Resource는 운송 자원을 의미한다. Agent에 작업이 주어졌을 때 혹은 작업 수행 중에 새로운 작업이 추가 혹은 제거되었을 경우에 각각의 Agent 별로 Genetic Algorithm을 사용하여 가장 적절한 Resource를 할당한다.



[그림 1] Problem Description

다음은 본 연구에서 사용되는 문제정의를 위한 표기법을 나타낸다.

- $A = \{A_1, A_2, \dots, A_p\}$: 물류 회사의 집합
- $R = \{R_1, R_2, \dots, R_q\}$: 운송 자원의 집합
- $U(A_i)$: 전체 운송자원의 집합 중 A_i 가 사용할 수 있는 운송 자원의 집합
- $O_i = \{O_{i1}, O_{i2}, \dots, O_{ik}\}$: i 번째 물류 회사가 담당하는 작업의 집합
- $T_i = \{T_{i1}, T_{i2}, \dots, T_{ij}\}$: i 번째 물류 회사가 담당하는 작업 중 Time Window내의 작업의 집합
- $D(T_{ij})$: T_{ij} 의 예정 완료일
- $P(T_{ij}, R_k)$: A_i 가 R_k 로 T_{ij} 를 수행하는데 걸리는 예상 시간
- $C(T_{ij}, R_k)$: A_i 가 R_k 로 T_{ij} 를 수행하는데 드는 예상 비용
- $PR_i = \{(T_{ij}, T_{ik})\}$: i 번째 물류 회사가 담당하는 작업의 선행 관계의 집합 (T_{ij} 가 T_{ik} 보다 선행해야 한다.)
- $CA(A_i)$: A_i 의 총 작업 개수
- N_i : i 번째 물류 회사의 총 작업 개수
- $Py(T_{ij})$: T_{ij} 를 포기할 경우 발생하는 비용
- $I_{ij} = \infty$ (Due Date를 반드시 지켜야 하는 작업), or 1 (Otherwise)

문제 해결에 필요한 가정은 다음과 같다.

- ① 한 물류회사는 한번에 하나의 동시작업만 수행할 수 있다.
- ② 모든 작업의 운송 부피는 1단위로 동일하고 하나의 운송자원이 사용된다. (2단위 이상일 경우 1단위의 여러 작업으로 나눈다.)
- ③ 물류 회사 간에 공유하는 운송 수단이 존재하며, 한 운송자원을 동시에 두 개 이상의 회사가 사용할 수 없다.
- ④ 운송수단이 사용되고 있지 않을 경우 언제든지 그 운송수단을 사용할 수 있다.

3. Proposed Algorithm

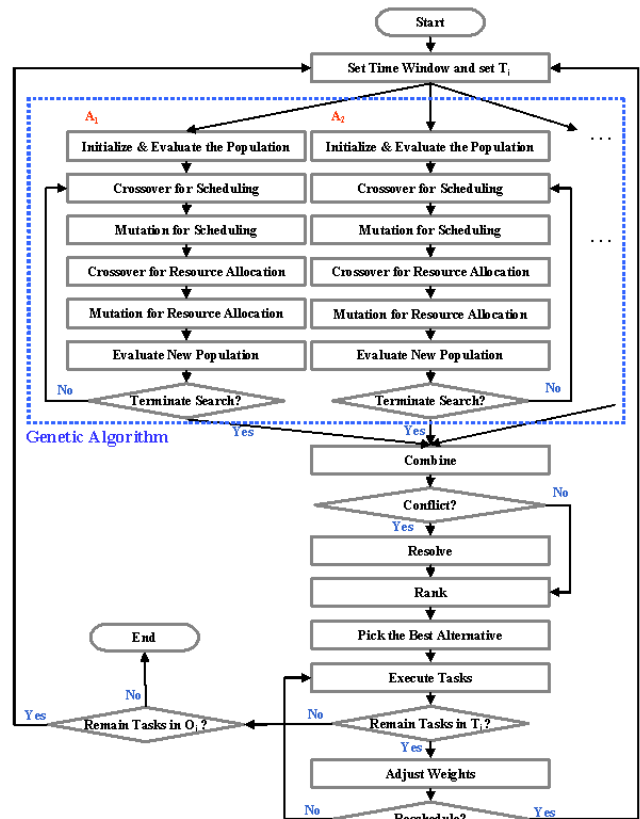
3.1. Genetic Algorithm

문제의 해결을 위하여 본 연구에서는 Genetic Algorithm을 사용하였다. 과거의 최적화 문제에서는 가능한 경우를 모두 고려하는 전통적인 Exhaustive Search Technique이 많이 사용되었다. 그러나 문제의 규모가 커질 때는 이 방법이 효과적이나 문제의 크기가 커질 경우에는 Computation Time과 사용 Memory의 기하급수적 증가로 인한 단점을 보인다[5]. 이를 해결하기 위한 Artificial Intelligence Technique중의 하나가 Genetic Algorithm이다. Genetic Algorithm은 비교적 여러 분야에 쉽게 적용될 수 있다는 점과 빠른 시간 내에 최적에 근사한 해에 접근할 수 있다는 점 등의 장점을 가지고 있다.

3.2. Overall Process

본 연구는 [그림 2]와 같은 Process를 따른다. 물류 회사별로 독립적으로 각 회사의 정책을 반영한 Genetic Algorithm을 이용하여 우수한 해를 도출한 후에 여러 회사의 결과를 종합하여 우수한 Performance를 낼 수 있는 해를 최종적으로 선택한다. 먼저, 회사별 Local Optimization을 위하여 우선 초기 Population을 생성하고, 각각의 물류 회사별로 Scheduling과 Resource Allocation에 관련된 Crossover와 Mutation을 종료 조건을 만족할 때까지 실시한다. 이로부터 생성된 각 물류 회사별 Population중에서 Fitness Value가 높은 상위 몇 개를 선정하고, 물류 회사들 사이의 결합을 통해 종합적인 Schedule을 만들어 낸다. 이때 자원 이용에 있어서 충돌이 발생할 경우에는 이를 해결하고, 최종적으로 Fitness Value가 가장 좋은 Best Alternative를 선정함으로써 Process는 종료된다. 여기서 Scheduling은 작업의 시작 시각과 종료 시각을 결정하는 것이고, Resource Allocation은 각 작업을 수행하는데 적절한 운송 수단을 할당하는 것이다. 선정된 Scheduling과

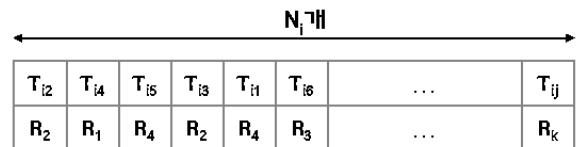
Allocation에 따라 작업을 수행하면서 Rescheduling 조건을 계속적으로 검사하다가 Rescheduling이 필요하면 남은 작업에 대해서 다시 해를 도출한다.



[그림 2] 전반적 문제해결 과정

3.3. Encoding Scheme

Genetic Algorithm에서 사용될 Chromosome의 구조는 [그림 3]과 같다. 총 셀의 개수는 i 번째 물류 회사 작업 개수의 2배인 $2N_i$ 개이고 각 셀에는 작업 또는 운송 자원이 들어간다. 1행 n 열의 작업을 수행하는데 사용되는 운송 자원은 2행 n 열에 들어있다. 예를 들어, [그림 3]에서 i 번째 물류 회사의 작업 T_{i2} (1행 1열)를 수행하기 위하여 운송 자원 R_2 (2행 1열)를 사용한다.



[그림 3] Chromosome의 구조

3.4. Population Initialization

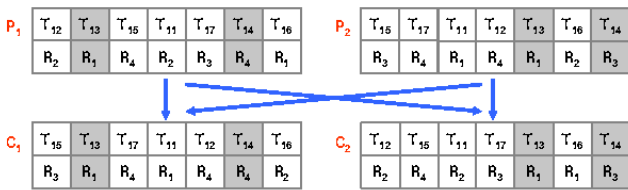
1행 1열에서 1행 N_i 열까지는 선행 관계를 고려하여 모든 작업을 할당한다. Feasible Population을 생성하기 위하여 Lee(2003)[3]가 제시한 다음과 같은 방법을 사용한다.

- Step 1.** 길이 N_i 의 빈 배열을 생성한다.
- Step 2.** 1에서 N_i 사이의 임의의 수 c 를 선택한다.
- Step 3.** 만약 작업 T_{ij} 에 해당하는 c 가 현재의 배열에 포함되어있지 않고, 동시에 그 작업보다 선행해야 하는 모든 작업이 배열에 포함되어 있다면 배열의 끝에 c 를 추가한다.
- Step 4.** 모든 작업이 배열에 추가되었다면 배열을 반환하고, 그렇지 않다면 Step 2로 되돌아간다.

2행 1열에서 2행 N_i 열까지는 각각 1행 1열에서 1행 N_i 열에 해당하는 물류 회사의 작업에 사용 가능한 운송 자원을 임의로 할당한다.

3.5. Genetic Operators for Scheduling

Genetic Algorithm에서 전형적으로 사용되는 두 가지의 Genetic Operator는 Crossover와 Mutation이다. Crossover Operator는 전체 Population으로부터 서로 다른 한 쌍의 Parent Chromosome(P_1, P_2)을 선택하여 이들의 교배를 통하여 새로운 Child Chromosome(C_1, C_2)을 만들어 낸다. 본 연구의 Scheduling에서 사용된 Crossover Operator는 다음과 같다. 우선 P_1 과 P_2 는 룰렛 선택법(Roulette Selection)에 따라 선택한다. 선택된 P_1 에서 임의의 작업을 선정하여 그와 선행 관계에 있는 작업을 모두 표시한다. 또한 표시된 작업과 선행 관계에 있는 작업 또한 모두 표시한다. 더 이상 표시할 작업이 없을 때까지 이러한 과정을 반복하여 표시된 작업들을 C_1 의 같은 위치로 복사한다. 그 외의 작업들은 P_2 로부터 C_1 에 차례로 복사한다. P_2 에서 C_1 으로 복사되고 남은 작업들은 C_2 의 같은 위치로 복사되고 그 외의 작업들은 P_1 으로부터 C_2 에 차례로 복사한다. 작업들이 이동할 때 그에 해당하는 운송 자원들도 함께 이동된다. [그림 4]는 Crossover의 예를 보여준다.



[그림 4] Scheduling에서 Crossover의 예

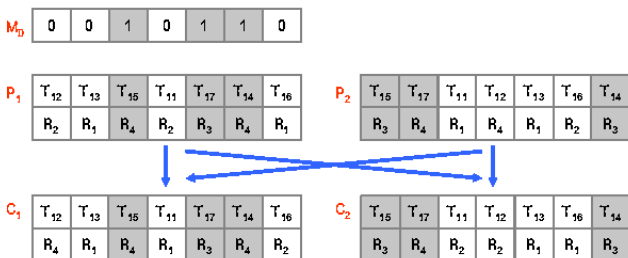
Mutation Operator는 특정 Chromosome에서 임의로 선택된 하나 이상의 셀을 변형시키는 것이다. 본 연구의 Scheduling에서 사용된 Mutation Operator는 다음과 같다. 우선 임의의 작업을 선정한다. 그리고 그와 선행 관계에 있는 작업들을 표시한다. 또한 표시된 작업과 선행 관계에 있는 작업 또한 모두 표시한다. 더 이상 표시할 작업이 없을 때까지 이러한 과정을 반복한 후 표시된 작업 중에 선택된 작업의 왼쪽에서 가장 가까이 있는 작업과 오른쪽에서 가장 가까이 있는 작업을 찾고, 그 사이에 있는 임의의 작업을 선정하여 그것과 순서를 바꾼다. 작업의 순서가 바뀔 때 그에 해당하는 운송 자원의 순서도 함께 바뀐다. [그림 5]는 Mutation의 예를 보여준다.



[그림 5] Scheduling에서의 Mutation의 예

3.6. Genetic Operators for Resource Allocation

Resource Allocation에서의 Crossover는 다음과 같다. 우선 3.5절의 Scheduling에서 생성된 C_1 과 C_2 를 각각 P_1 과 P_2 로 사용한다. 또한 Masking을 위한 배열 M_p 를 만들어 각 셀에 0 또는 1을 임의로 생성하여 할당한다. P_1 을 기준으로 1이 할당된 셀에 해당하는 Task의 Resource는 P_1 에서, 그 외의 Resource는 P_2 에서 물려받아 C_1 을 만든다. 마찬가지로 P_2 을 기준으로 1이 할당된 셀에 해당하는 Task의 Resource는 P_2 에서, 그 외의 Resource는 P_1 에서 물려받아 C_2 을 만든다. [그림 6]은 이러한 Crossover의 예를 보여준다.



[그림 6] Resource Allocation에서의 Crossover의 예

Resource Allocation에서의 Mutation은 다음과 같다. 임의의 작업을 선정하여 작업에 해당하는 운송 자원을 변형시킨다. 이때, 변경되는 운송 자원은 각 물류 회사에서 사용 가능한 자원으로 한정된다. [그림 7]은 Mutation의 예를 보여준다.



[그림 7] Resource Allocation에서의 Mutation의 예

이상과 같은 Crossover와 Mutation을 통해 정해진 선행 관계를 지키며 가능한 다양한 경우를 나타내는 Chromosome을 생성할 수 있다. 즉, 작업의 순서도 바꿀 수 있을 뿐 아니라 각 작업을 수행할 운송 자원도 바꿀 수 있다.

3.7. Fitness Evaluation

이 연구에서 사용하게 될 적합도 함수에는 운송 비용과 운송 지연에 관련된 Factor가 포함된다. 적합도 함수는 다음과 같다.

$$FV = w_1 \times (CostFactor) + w_2 \times (DelayFactor)$$

여기서 w_1 과 w_2 는 각각 Cost Factor(CF)와 Delay Factor(DF)의 Weight 값으로 $w_1 + w_2 = 1$ ($w_1, w_2 \geq 0$)을 만족한다. w_1 과 w_2 의 초기값은 Cost와 Delay의 중요도에 따라 각 물류 회사별로 결정된다. 이렇게 도출된 적합도 함수를 최소로 만들어 주는 Genetic Algorithm을 개발하는 것이 본 연구에서의 목적이다.

Cost Factor(CF)는 다음과 같이 계산된다.

$$CF = \sum_{i=1}^p \sum_{j=1}^{CA(A_i)} C(T_{ij}, R_k)$$

여기서 R_k 는 T_{ij} 에 해당하는 운송 자원이고 p 는 물류 회사의 개수이다.

Delay Factor를 계산하기 위해서는 각 작업의 시작 시간과 종료 시간을 표현해야 한다. 작업 T_{ij} 의 시작 시간(ST)과 종료 시간(FT)은 각각 다음과 같이 정의된다.

$$ST(T_{ij}) = \max\{T_{ij} \text{보다 선행 관계가 빠른 작업들의 종료 시간}, T_{ij} \text{를 수행할 운송 자원의 이용 가능한 가장 빠른 시간}\}$$

$$FT(T_{ij}) = ST(T_{ij}) + P(T_{ij}, R_k)$$

이를 종합한 Delay Factor는 다음과 같이 계산된다.

$$DF = \sum_{i=1}^p \sum_{j=1}^{CA(A_i)} [I_{ij} \times \max\{(FT(T_{ij}) - D(T_{ij})), 0\}]$$

이상과 같은 CF와 DF는 Normalization을 통하여 비교 가능한 Scale의 값으로 바꾼다.

3.8. Combination

이상과 같은 방법으로 생성된 각 회사별 Chromosome을 하나의 Chromosome으로 결합하는 과정이 Combination이다. 지금까지 다룬 Genetic Algorithm을 이용하여 생성된 물류 회사의 Population 중에서 각각 Fitness를 기준으로 상위 n 개의 Chromosome을 선정하여 결합한다. 이 경우 모든 작업을 결합할 수 있는 경우의 수는 n^p 개(p : 물류 회사의 총 개수)이므로 n^p 개의 결합된 Chromosome이 생성된다. 결합된 모든 Chromosome에 대하여 간트 차트를 작성한다.

그 결과 두 개 이상의 물류 회사가 동일한 시간에 동일한 운송 자원을 사용하도록 할당된 경우 Conflict가 발생한다. 만약 운송 자원을 사용하는데 있어서 Conflict가 발생하지 않는다면 3.7절에서 제시된 Fitness 도출 방법에 따라 모든 회사의 작업을 고려한 FV를 도출하고, Conflict가 발생하는 경우에는 Conflict가 발생하는 작업을 포기하여 제거한 후에 FV를 계산한다. 이때는 작업 포기시의 Penalty에 해당하는 비용 $P_y(T_{ij})$ 을 고려하여 비용 증가가 많은 쪽으로 운송 자원

을 할당하고 비용증가가 적은 작업을 포기하고 제거한다. Conflict가 발생하는 Chromosome의 Cost Factor에는 포기되는 작업의 $P_y(T_{ij})$ 를 더해준다. 이렇게 계산된 FV를 기준으로 하여 Ranking을 한 다음 FV가 가장 낮은 해를 최종적으로 Best Alternative로 선정한다.

3.9. Time Window

본 연구에서 제시한 Algorithm을 수행할 때 할당된 모든 작업을 고려하는 것은 비효율적이다. 고려해야 하는 경우의 수가 많고 실시간으로 Rescheduling과 Reallocation을 해야 하므로 Search Space를 줄임으로써 Computational Time과 사용 Memory를 줄이는 것이 필요하다. 따라서 현재를 기준으로 일정 시간 범위 내의 작업에 대해서만 Scheduling과 Resource Allocation을 하는 Time Window 접근법[6]을 적용한다. 고려 대상이 되는 작업의 집합 T_i 를 다음과 같은 방법으로 도출하여 Genetic Algorithm을 수행한다.

- Step 1. T_i 를 공집합으로 초기화한다.
- Step 2. 각 물류 회사별로 O_i 에 포함된 모든 작업 T_r 에 대하여 Due Date를 기준으로 정렬한다.
- Step 3. Time Window 영역 내에 Due Date가 위치하는 작업들을 선정한다.
- Step 4. 선정된 작업들을 T_i 에 포함시킨다.

4. Rescheduling and Reallocation

다음과 같은 경우에 작업을 Rescheduling하고 각 작업의 운송 자원을 Reallocation한다.

- ① Time Window내에 작업이 추가되었을 때
- ② 작업의 예측된 완료 시간과 실제 완료 시간이 일정 값 이상의 차이를 보일 때
- ③ 운송 자원이 추가되었을 때, 혹은 고장, 사고 등으로 사용할 수 없게 되었을 때

Rescheduling과 Reallocation을 할 때 Chromosome의 셀의 개수가 변화한다. 작업이 추가되는 경우 작업 한 개당 두 개(작업과 그에 해당하는 운송 자원)의 셀이 늘어나고 진행 중인 작업과 완료된 작업을 제외 할 때는 작업 한 개당 두개의 셀이 감소한다. 작업 수의 변화가 없고 운송 자원만 변화하는 경우는 셀의 개수가 유지된다. 셀의 개수를 변화시킨 후에 진행 중인 작업과 완료된 작업을 제외한 나머지 작업에 대하여 3절에서 설명한 방법으로 Rescheduling하고 각 작업의 운송 자원을 Reallocation한다. 작업의 예정 종료시간과 실제 종료시간에는 오차가 존재하는데 이 오차는 다수의 작업이 진행되는 과정에서 누적되어 증가함으로써 전반적인 시스템의 성능 저하를 유발할 뿐 아니라 후행하는 작업과정에서 예정에 없던 Conflict를 발생시킬 수 있다. 따라서 모든 작업이 완료될 때 마다 ②의 경우처럼 예정 완료시간과 실제 완료시간을 비교하여 기준값 이상으로 오차가 커질 경우 Time Window를 다시 설정하고 Rescheduling과 Reallocation과정을 통해 일정을 재조정해야한다.

5. Weight Adjustment

Cost Factor(CF)와 Delay Factor(DF)의 Weight인 w_1 과 w_2 는 각 회사의 정책을 고려한 요소로써, 정책이 변경에 따라서 또는 작업 수행의 효율 향상을 위해서 w_1 과 w_2 를 조절해야 할 필요가 있다. Genetic Algorithm을 이용한 Scheduling과 Allocation을 수행시 본 연구에서는 다음과 같은 w_1 과 w_2 의 조절 정책을 사용한다.

$$IF \frac{R(CF)/S(CF)}{R(DF)/S(DF)} \geq \alpha, THEN w_1 \leftarrow w_1 \times (1 + \beta), w_2 \leftarrow w_2 - w_1 \times \beta$$

$$IF \frac{R(DF)/S(DF)}{R(CF)/S(CF)} \geq \alpha, THEN w_2 \leftarrow w_2 \times (1 + \beta), w_1 \leftarrow w_1 - w_2 \times \beta$$

즉, CF의 값이 DF값에 비하여 상대적으로 기준치 이상으로 높은 경우 CF의 Weight를 높여주고, 반대로 DF의 값이 상대적으로 기준치 이상으로 높은 경우 DF의 Weight를 높여주는 정책이다. 여기서 $R(CF)$, $S(CF)$, $R(DF)$, $S(DF)$ 는 각각 Cost Factor의 실제값, Cost Factor의 기준값, Delay Factor의 실제값, Delay Factor의 기준값을 의미한다. 기준값이란 허용 가능한 대략적인 값으로, 미리 예측하여 정해진 상수값이다. 또한 $\alpha (\geq 1)$ 는 Weight의 조정 여부를 결정하는 기준점이 되는 Threshold이며, $\beta (\geq 0)$ 는 Weight의 조정 정도를 나타내는 Adjustment Factor이다. 각 작업이 끝날 때 마다 Weight 조절 조건을 검사하고 조건을 만족할 경우 Weight를 수정한다. 이러한 Weight의 Dynamic한 변화를 통해 보다 효과적인 Rescheduling과 Reallocation을 할 수 있다.

5. Conclusions

이 연구에서는 운송비용과 납기일, 운송자원의 가용성, 작업의 선행관계, 납기일에 따른 작업의 우선 순위 등을 고려하여 작업 순서를 정하고 운송 자원을 할당하는 방법에 대하여 모색해 보았다. 또한 시스템에 변화가 발생하였을 때 운송 자원을 재할당하는 정책에 대해서도 살펴보았다. 규모가 큰 문제에 대하여 실시간 정책에 사용가능한 유연성과 비교적 높은 효율성을 보이는 Genetic Algorithm을 사용하여 Algorithm을 개발하였으며, 이를 위한 효과적인 Chromosome의 구조와 Genetic Operator, 그리고 Fitness Value를 제시하였다. 이 연구를 통하여 물류 회사의 각 작업에 가장 적절한 운송 자원을 효율적으로 할당할 수 있을 것이다.

이러한 Methodology를 사용하여 실제로 이용 가능한 시스템을 구현하고 효율성을 검증하며, 특히 작업간의 Conflict를 효과적으로 해결하는 방법론에 관한 추후 연구가 필요하다.

Acknowledgement

이 논문은 2003년도 한국학술진흥재단의 지원에 의하여 연구되었음 (KRF-2003-041-D00622).

참고문헌

- [1] Cargonews, <http://www.cargonews.co.kr/gisa/200304/030421-1.htm>, 21 April, 2003
- [2] Baita, F., Pesenti, R., Ukovich, W. and Favaretto, D. (2000), A comparison of different solution approaches to the vehicle scheduling problem in a practical case, *Computers & Operations Research*, Vol. 27, 1240-1269.
- [3] Lee, K. M. and Lee, J. H. (2003), Coordinated Collaboration of Multiagent Systems Based on Genetic Algorithms, *Proceedings of PRIMA 2003*, 145-157.
- [4] David, T. and Pratyush, S. (1999), Distributed Task Scheduling and Allocation using Genetic Algorithms, *Computers & Industrial Engineering*, Vol. 37, 47-50.
- [5] Cha, Y. P. and Jung, M. Y. (1999), 3D part packing : a genetic algorithm based approach, *Proceedings of The 1999 KIIIE Fall Conference*, 49-52.
- [6] Solomon, M. M. and Desrosiers, J.(1998), Time window constrained routing and scheduling problems, *Transport. Sci.*, Vol. 22, 1-13.