

오픈마켓 포트폴리오 관리를 위한 프로젝트 우선순위결정

이 용 희[†] · 이 건 호^{††}

요 약

최근 오픈마켓 시장은 선후발 업체간의 경쟁력 강화를 목적으로 다양한 프로젝트가 진행되면서, 프로젝트 포트폴리오 관리의 중요성도 함께 높아지고 있다. 프로젝트 포트폴리오 관리는 자원 제약환경에서 대기된 다수의 프로젝트 중 전략적 목적에 부합되는 프로젝트를 선택하여 이익을 최대화하는 중요한 의사결정 활동이다. 본 논문에서는 프로젝트 포트폴리오 관리의 연구 동향을 소개하고 프로젝트들의 가치, 성공가능성, 프로젝트 유사도를 고려한 포트폴리오 선정방법과 자원 제약조건에서 포트폴리오 평가점수의 최대화를 목적으로 한 포트폴리오 우선순위결정 문제를 모형화하고 유전자 알고리즘을 이용하여 이를 해결하는 시스템으로 제시하고 있다. 구현된 포트폴리오 시스템을 검증하기 위하여 위해 한국의 E-기업 오픈마켓의 기존 우선순위, 개발인력 할당 등과 비교분석하고 있다. 제시된 시스템을 사용함으로써 가치있는 프로젝트와 성공 가능성이 높은 프로젝트를 우선적으로 수행하게 하게 하고, 유사한 프로젝트들을 연속하여 개발하게 하여 개발업무에 연속성을 부여하는 결과를 얻을 수 있다. 이는 궁극적으로 인력활용도를 높이고 개발업무를 효율화하여 프로젝트들의 위험(risk)을 줄이고, 성공 가능성을 높이는데 기여하고 있다.

키워드 : 오픈마켓, 포트폴리오, 유전자 알고리즘, 프로젝트 유사도

Ordering of Project priorities For Open Market Portfolio

Yong-Hee Lee[†] · Gun-Ho Lee^{††}

ABSTRACT

In the recent years, a variety of projects have been conducted in order to enhance competitiveness of leading businesses and their followers in the market. Accordingly, the importance of project portfolio management has risen in the open market industry. Project portfolio management refers to crucial decision-making processes which aim to maximize benefits by selecting projects most suitable for a strategic objective among multiple projects with limited resources. In this study, the trend of project portfolio management studies is introduced. The study also presents a mathematical model of the problem, which aims at maximizing project values, possibility, and similarity between projects in the limited resources. We use the genetic algorithm to obtain the priority orders of projects. In order to verify this study, we compare the results of this study and the existing schedules of the E-open market in South Korea. This study ultimately reduces project risks, improves efficiency of development and continuity of tasks by properly ordering projects and assigning developers to the projects.

Keywords : Open Market, Project Portfolio, Priority, Project Management

1. 서 론

최근 오픈마켓 시장에서는 선후발 업체간 경쟁이 과열되면서 경쟁력 강화를 목적으로, 다양한 개발 프로젝트가 동시다발적으로 진행되고 있다. 이러한 환경에서 프로젝트의 중요성이 높아짐에 따라 프로젝트의 우선순위 결정과 효율적인 스케줄링을 위한 프로젝트 포트폴리오(Portfolio) 관리

의 중요성도 함께 높아지고 있다. 포트폴리오 관리는 자원 제약환경에서 다수의 프로젝트 중 전략적 목적에 부합되는 프로젝트를 선택하여 수행하는 중요한 의사결정 활동이다 [1]. 그렇지만 포트폴리오 관리에는 많은 불확실한 요소가 내포되어 올바른 판단에 어려움이 있으며, 전문가에 의한 경험적 판단이나 프로젝트를 요청하는 팀의 요구 강도에 의하여 프로젝트의 우선순위가 결정되고 있다. 이러한 문제점으로 인해 비교적 가치가 작은 프로젝트가 우선적으로 진행되어 자원 낭비와 수익이 감소하게 되며, 결과적으로 시장 경쟁력 약화로 이어지게 된다.

위와 같은 상황에서 본 논문은 오픈마켓을 위한 효율적인 프로젝트 포트폴리오 관리방법을 제시하는 것으로 연구의

※ 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음.

† 중신회원 : 숭실대학교 대학원 산업정보시스템공학과

†† 정 회 원 : 숭실대학교 산업정보시스템공학과 교수

논문접수 : 2011년 2월 9일

수정일 : 1차 2011년 4월 11일

심사완료 : 2011년 5월 9일

모델은 한국 최대 오픈마켓 중의 하나인 E-기업을 대상으로 하고 있다.

본 논문에서는 프로젝트 포트폴리오를 위해 프로젝트들의 가치, 성공가능성, 프로젝트 유사도의 최대화를 목적으로하는 프로젝트 우선 순위 결정, 개발자의 할당을 해결하는 수학적 모형을 제시하고 있다. 하지만 자원의 할당과 우선 순위 결정의 문제는 유한 시간 내에 최적해를 구하기 어려운 문제이므로 유전자 알고리즘을 이용하여 해결하고자 한다. 또한 실무적으로 사용 가능한 시스템을 구현하여 제시하고 있다.

구현된 포트폴리오 시스템을 검증하기 위하여 E-사에서 수행한 프로젝트의 포트폴리오와 동일 문제를 본 논문의 시스템에 적용한 우선순위, 개발인력 할당 등의 결과와 비교 분석하고 있다. 본 논문에서 인력의 활용도를 개선하고 전체 프로젝트의 가치가 증가되고 성공가능성도 증가되는 효과를 볼 수 있다. 가치있는 프로젝트와 성공 가능성이 높은 프로젝트를 우선적으로 수행하게 하여 유사한 프로젝트들을 연속하여 개발하게 함으로써 개발업무에 연속성을 부여하는 결과를 얻을 수 있다. 이는 궁극적으로 인력활용도를 높이고 개발업무를 효율화하여 프로젝트들의 가치를 극대화하고 성공 가능성을 높이는데 기여하고 있다.

본 논문의 2장에서는 포트폴리오 관리, 스케줄링, 프로젝트 유사도에 관한 연구동향을 고찰하고, 3장에서는 해결하고자 하는 문제의 제시, 프로젝트가치, 성공가능성, 자원 제약, 유사도를 각각 정의하고 수학적인 모형을 제시하고 있다. 4장에서는 제시한 모델을 유전자알고리즘을 이용하여 시스템으로 구현하며, 5장에서는 시스템성과 스케줄링 결과를 평가하고, 6장에서는 결론에 대해 언급하였다.

2. 관련 연구

본 논문에서 관련연구 조사는 프로젝트 우선순위 결정 요인을 선택하기 위한 포트폴리오 관리에 대한 연구, 자원 제약 조건에서 우선순위를 고려하여 프로젝트 실행 계획을 세우기 위한 스케줄링에 관한 연구, 그리고 유사 프로젝트를 연속 진행시켜 성공가능성을 높이기 위한 프로젝트 유사도 측정에 관한 세가지의 연구로 구분하여 조사하였다.

첫째, 포트폴리오 관리 연구로는 Ringuest[2]의 휴리스틱 기법을 이용하여 포트폴리오 선정과 기존 포트폴리오에 신규 프로젝트를 추가하였을 때 선택방법에 대한 연구가 있으며, Ghasemzadeh[3]는 계층적분석기법(AHP), 선형목표계획법(LGP)을 사용하여 포트폴리오 선정에 필요한 변수들을 고려하여 포트폴리오 선정모형을 제시하였다. Stummer[4]는 휴리스틱 기법으로 선정된 프로젝트에 대하여 포트폴리오 선정에 관련한 검증 방법을 제시하였으며, Chen과 Askin[5]은 자원 할당을 위한 프로젝트 우선순위를 기준으로 한 포트폴리오 선정 방법을 제시하였다.

둘째, 프로젝트 스케줄링 관련 연구는 활동(Activities)간의 연결관계를 고려한 순서결정의 해를 찾는 문제로 시작되

어, 자원제약조건 프로젝트 스케줄링(RCPSP: Resource-Constrained Project Scheduling Problem) 연구로 발전되었다. 이 후 RCPSP가 NP-complete으로 인식되면서 다양한 기법으로 연구가 진행되고 있으며[6], 그 중 유전자 알고리즘을 이용한 연구가 활발하게 진행되고 있다. Gonçalves[7]는 자원제약조건에서 멀티프로젝트 문제를 해결하기 위해서 프로젝트의 대기시간, 배포시간, 작업시간을 염색체 정보로 유전자 연산을 수행하여 우선순위를 결정하였고, Harper[8]는 병렬프로젝트 환경에서 한정된 인력으로 작업의 시작과 종료시간을 고려하여, 효율적인 인력 배정을 유전자 알고리즘을 사용하여 해결하였으며, Alba[9]는 소프트웨어 프로젝트 스케줄을 작성을 위해서 재무적인 요소를 염색체로 활용하여 유전자연산을 수행하여 해결하였다.

셋째, 프로젝트 유사도 측정 연구는 Shepper의 프로젝트의 구축 비용을 산정하기 위하여 과거의 진행된 프로젝트와 유사도를 측정한 연구[10]와 Sadahiro의 재사용을 목적으로 두 개의 소프트웨어의 유사도를 측정한 연구[11]가 진행되었다. 유사도 측정 기법으로는 Musilek 와 Idri의 프로젝트의 특성을 측정하여 변수간의 유사도를 퍼지로지(Fuzzy Logic)을 사용하여 측정한 연구[12,13]와 Kadoda와 Bisio의 축적된 프로젝트 데이터를 분석한 사례기반 연구[14,15]와 Robert의 구축된 프로그램 모듈의 유사성을 분석하여 유사도를 측정 한 연구가 있다[16].

위와 같은 유전자 알고리즘을 활용한 연구 동향에서, 기존 연구의 목적이 주로 자원제약 조건에서 자원활용의 최대화를 목적으로 포트폴리오 관리가 진행되어, 프로젝트 가치와 구현 가능성을 프로젝트 스케줄에 반영하지 않고, 유사 프로젝트의 연속 진행 시 수행 비용이 감소됨을 고려하지 못한 단점이 있다.

3. 프로젝트 포트폴리오 관리 모형

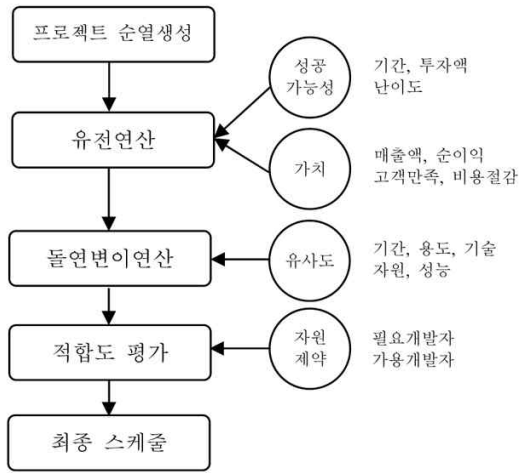
프로젝트 포트폴리오관리의 고려요소는 구현 후 얻을 수 있는 프로젝트의 가치, 규모와 난이도를 고려한 성공가능성, 한정된 자원을 효율적으로 이용할 수 있는 자원활용이며, 유사 프로젝트를 연속 진행 시 수행 기간을 단축할 수 있는 유사도를 측정하여 성공가능성을 보정한다. 다음 (그림 1)은 프로젝트 포트폴리오의 고려요소를 적용한 프로젝트 포트폴리오 선정 프로세스이다.

본 논문에서는 프로젝트 포트폴리오 모형을 개발하기 위하여 기존 자원제약 조건에서 다중 프로젝트 스케줄링 연구모델[7,8]에 우선순위 결정요인과 유사프로젝트에 대한 제약 조건을 보완하여, 다음과 같은 모형을 제시한다. 포트폴리오의 목적은 n개의 프로젝트를 대상으로 제약조건을 만족시키면서, 적합도(프로젝트 우선순위 점수)를 최대화하는 것이다.

-사용기호

i: 프로젝트 인덱스(i=1, ..., n)

j: 스케줄 기간 인덱스(j=1, ..., l)



(그림 1) 프로젝트 포트폴리오 선정 프로세스

- k: 개발자 인덱스(k=1, ..., m)
- h_i: 프로젝트 i의 투입개발자 수
- H: 가용한 최대 개발자 수
- t_i: 프로젝트 i의 수행 시간
- T: 가용한 최대 기간
- v_i: 프로젝트 i의 가치
- q_i: 프로젝트 i의 성공가능성
- w_i: 프로젝트 i의 가중치(weight)
- S_{ig}: 프로젝트 i와 g간의 유사도

-결정변수

$$u_{ig} = \begin{cases} 1: \text{만약 프로젝트 } i \text{와 } g \text{가 동시에 선택될 경우} \\ 0: \text{그렇지 않으면} \end{cases}$$

$$x_{ij} = \begin{cases} 1: \text{프로젝트 } i \text{가 } j \text{시간에 할당되는 경우} \\ 0: \text{그렇지 않으면} \end{cases}$$

$$y_{ik} = \begin{cases} 1: \text{인력 } k \text{가 프로젝트 } i \text{에 할당되는 경우} \\ 0: \text{그렇지 않으면} \end{cases}$$

$$z_{jk} = \begin{cases} 1: \text{인력 } k \text{가 } j \text{시간에 할당되는 경우} \\ 0: \text{그렇지 않으면} \end{cases}$$

-목적함수

$$\max \sum_{j=1}^l \sum_{i=1}^n x_{ij}(v_i + q_i) + \sum_{g=1}^n \sum_{i=1}^n u_{ig} S_{ig}$$

-제약조건

$$\sum_{j=1}^l \sum_{i=1}^n x_{ij} t_i \leq T \tag{1}$$

$$\sum_{k=1}^m \sum_{i=1}^n y_{ik} h_i \leq H \tag{2}$$

$$\sum_{j=1}^l z_{jk} = 1 \quad (k = 1, \dots, m) \tag{3}$$

$$\sum_{k=1}^m y_{ik} = h_i \quad \text{for } i = 1, \dots, n \tag{4}$$

$$\sum_{i=1}^n x_{ij} + \sum_{g=1}^n x_{g(j+1)} \leq 2u_{ig} \quad (j=1, \dots, l) \tag{5}$$

$$x_{ij} + y_{jk} \geq 2z_{ik} \quad (i = 1, \dots, n, j = 1, \dots, l, k = 1, \dots, m) \tag{6}$$

$$x_{ij}, y_{jk} = 0 \text{ or } 1, z_{ik} = 0 \text{ or } 1 \tag{7}$$

제약조건 (1)은 프로젝트의 수행기간의 합은 스케줄 대상 기간을 초과 할 수 없으며, (2)는 모든 프로젝트에 할당된 개발인력의 합은 최대 가용 개발인력을 초과하지 않도록 한다. (3)은 개발인력을 동일 시간대에서는 하나의 프로젝트에 만 참여할 수 있도록 하며, (4)는 각 프로젝트의 개발인력은 필요한 최소인력으로 구성되도록 한다. (5)는 두 프로젝트 i, g가 연속해서 진행되는 경우의 조건이며, (6)은 선택된 프로젝트, 선택된 기간, 할당된 개발 인력에 대한 변수들의 일관성을 부여한다. (7)은 결정변수의 선택이 가능 하도록 한다.

3.1 프로젝트 가치의 평가

프로젝트 완료 시 발생하는 가치는 측정하기 어려운 무형적인 요소가 많다. ISACA(Information Systems Audit and Control Association)는 프로젝트 우선순위의 결정요인으로 ROI, 사업가치, 사용기술, 법적규제, 시장 적시성으로 구분하여, 요인 별로 가중치를 부여한 평가 매트릭스를 제시하고 있다. 본 논문에서는 프로젝트의 가치를 용이하게 산출할 수 있도록, ISACA에서 제시한 결정요인과 E기업에서 사용하는 프로젝트 우선순위 평가표, 그리고 관련 업계 종사자들의 의견을 수렴하여 우선순위 결정요인을 도출하고, 프로젝트의 특성 별로 가중치를 부여하는 평가표를 작성하여 활

<표 1> 프로젝트 가치 산출 평가표

가중치	매출액	순이익	고객만족	비용절감	합계
프로젝트	3	3	2	2	
P1	7	3	1	3	35
P2	4	4	10	4	48
P3	2	6	4	5	37
P4	7	7	5	5	57
P5	4	4	7	7	45
P6	9	6	8	8	69
P7	3	4	4	9	38
P8	3	4	9	4	43
P9	6	5	4	5	46
P10	10	7	3	9	66

용한다. 프로젝트 가치를 산출하는 요소는 외형적 성장을 위한 매출액, 프로젝트 완료로 발생하는 순이익, 프로젝트를 통해 유발되는 고객만족, 프로세스개선과 전산화를 통한 비용절감이며 각 항목별로 1~10의 점수를 기입하고 항목별 중요도를 고려하여 가중치를 부여한다. 다음 <표 1>은 작성된 프로젝트 가치 산출 평가표이다[17].

프로젝트 가치 산출 평가표에서 프로젝트*i*에 대한 가치의 속성의 합계(v_i)는 다음과 같은 식으로 계산한다.

p_{ik} : 프로젝트 *i*의 *k*번째 가치 속성 점수($k=1, \dots, m$)

w_{ik} : 프로젝트 *i*의 *k*번째 가치 속성의 가중치

$$v_i = \sum_{k=1}^4 (p_{ik} w_{ik}) \quad (i=1, \dots, n)$$

3.2 프로젝트 성공가능성의 평가

프로젝트의 성공가능성을 산출하는 요소로는 프로젝트의 수행기간, 프로젝트 수행에 소요되는 투자액, 투입되는 인력, 구현의 난이도가 있으며 각 항목별 1~10의 점수를 기입하고 항목별 중요도를 고려하여 가중치를 부여한다. 다음 <표 2>는 작성된 프로젝트 성공가능성 산출 평가표이다.

<표 2> 프로젝트 성공가능성 산출 평가표

가중치	기간	투자액	투입인력	난이도	합계
프로젝트	1	2	1	2	
P1	3	3	2	9	29
P2	6	5	4	4	28
P3	7	6	5	6	36
P4	6	3	4	7	30
P5	8	5	5	4	31
P6	4	6	5	6	33
P7	5	3	6	2	21
P8	8	7	4	6	38
P9	8	3	6	4	28
P10	2	2	8	3	20

프로젝트 성공가능성 산출 평가표에서 프로젝트 *i*에 대한 성공가능성의 합계(q_i)는 다음과 같이 계산한다.

p_{ik} : 프로젝트 *i*의 *k*번째 성공가능성 속성 점수($k=1, \dots, n$)

w_{ik} : 프로젝트 *i*의 *k*번째 성공가능성 속성의 가중치

$$q_i = \sum_{k=1}^4 (p_{ik} w_{ik}) \quad (i=1, \dots, n)$$

프로젝트의 우선순위 점수는 v_i 와 q_i 의 합으로 결정된다.

3.3 필요 자원과 가용 자원

프로젝트 스케줄링은 한정된 자원 제약 안에서 병렬진행이 가능하다. 이러한 프로젝트 병렬진행을 계산하기 위해서는 프로젝트의 수행 기간, 투입인력 그리고 가용인력이 산정되어야 한다. 가용한 인력을 초과하지 않는 범위에서 병렬 진행하여, 가능한 많은 프로젝트가 수행 되어야 자원의 활용도가 높아지게 된다.

3.4 프로젝트 유사도 결정

다수의 프로젝트를 진행할 경우 유사 프로젝트를 연속 진행하면 분석/설계 시간 감소, 개발 소스의 재 사용으로 후행 프로젝트의 수행 비용이 감소한다[18]. 본 논문에서는 계층적군집분석(Hierarchical Clustering method)을 사용하여 유사도를 측정하며, 프로젝트의 유사도 측정 변수는 COCOMO에서 제시한 프로젝트 속성을 활용하였다[19]. 프로젝트의 속성별로 1~6의 점수로 평가하여 유사도 측정 데이터로 사용하고, 가중치를 계산하여 프로젝트 특성 점수를 산정하게 된다. 다음 <표 3>은 5개의 프로젝트의 특성 점수 표 예제이다.

<표 3> 프로젝트 특성 점수

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
RELY	2	3	4	5	1	2	3	4	5	2
DATA	2	3	2	3	4	5	2	5	2	4
CPLX	1	6	2	4	3	5	2	6	1	2
TIME	4	5	3	5	3	6	4	6	3	4
STOR	5	5	6	6	4	6	4	6	4	4
VIRT	2	2	3	3	4	4	5	4	5	2
TURN	3	4	5	3	2	3	5	2	3	4
ACAP	2	1	2	4	5	2	1	2	3	4
AEXP	2	3	4	1	4	2	4	5	1	2
PCAP	4	5	4	3	1	2	3	5	3	1
VEXP	1	2	4	1	2	2	4	2	3	4
LEXP	3	2	3	4	2	4	2	1	2	4
MODP	1	3	5	3	4	3	2	5	3	2
TOOL	2	3	5	3	5	5	3	2	3	4
SCED	2	3	4	5	5	1	3	2	5	3

프로젝트의 유사도를 결정하기 위해서 특성점수의 합을 기준으로 군집분석을 실행한다. 프로젝트의 유클리디안 거리(Euclidean distance)가 작을수록 두 프로젝트는 유사하며, 유클리디안 거리를 측정하기 위해서 다음과 같은 식을 사용한다.

a_{ip} : *i*프로젝트의 *p*특성점수

i: 프로젝트의 인덱스

p: 특성의 인덱스 ($p=1, \dots, r$)

$$\text{Similarity (sig)} = \sqrt{\sum_{p=1}^r (a_{ip} - a_{gp})^2}$$

서, 더 이상 결정되지 않은 유전자가 없을 때까지 연산을 진행하면 최종적 해는 8942513760이 된다. 다음 (그림 4)는 2차 주기교차연산을 나타낸다.

4.3 돌연변이 연산(Mutation Operations)

유전자 알고리즘에서 돌연변이 연산자는 교차에 의한 전역적인 탐색에서 제외된 유전자 안에서의 치환으로 해의 개선의 가능성을 생성하는 연산이다. 본 논문에서는 프로젝트 유사도를 돌연변이 연산으로 사용한다. 염색체 순열에서 유사프로젝트가 존재 할 경우 선행 프로젝트의 다음 프로젝트를 유사 프로젝트와 치환하게 되며, 이때 교환 돌연변이는 항상 유효한 경로를 생산해야 한다. 치환된 유전자는 적합도를 평가하여 해가 개선 되었을 경우 돌연변이를 유지하게 된다. 돌연변이 연산은 다음과 같은 로직으로 수행한다.

- S₁: 돌연변이 연산 전 순열
- S₂: 돌연변이 연산 후 순열
- V₁: S₁의 우선순위 점수
- V₂: S₂의 우선순위 점수

BEGIN

V₁ = Fitness (S₁)

S₂ = Mutation (S₁)

V₂ = Fitness (S₂)

IF EXIST(유사프로젝트) AND V₁ > V₂ THEN

V₁ = V₂

S₁ = S₂

END IF

END

4.4 적합도 평가(Fitness Check)

교배와 돌연변이 연산으로 생성된 염색체는 가용한 인력 안에서 유전자의 순열에 따라 병렬 진행 순서가 결정 된다. 다음은 결정된 순열을 이용한 프로젝트 스케줄링의 순서이다.

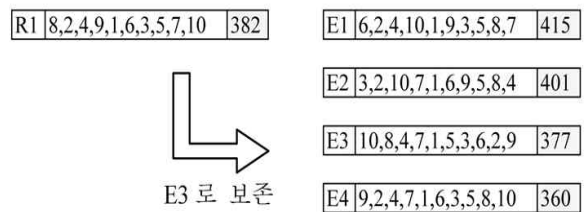
- Step1. 결과 테이블에서 R_y ≤ T_y를 만족하는 시작 좌표 x, y를 검색
 - R_y: 스케줄 대상 프로젝트의 필요 자원
 - T_y: 테이블에서 할당되지 않은 자원
- Step2. 시작 좌표부터 자원 × 시간의 값 만큼 칼럼에 프로젝트 번호를 입력
- Step3. 프로젝트의 개수만큼 Step1~Step2를 반복

다음 (그림 5)는 <표 5>의 염색체 정의 데이터를 대상으로 9,2,4,7,1,6,3,5,8,10의 순열을 가지는 염색체에 가용한 인력 10을 기준으로 프로젝트를 스케줄링 순서에 따라 실행한 예제이다.

우선순위가 결정된 프로젝트의 우선순위 점수를 적합도 기준으로 활용하며, 5개의 프로젝트를 선택하는 스케줄 조

기간 자원	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
1						2			7				6		5							10				
2																										
3																										
4	9																									
5																										
6																										
7																										
8						4			1	3																
9																										
10																										

(그림 5) 프로젝트 우선순위 적용 예제



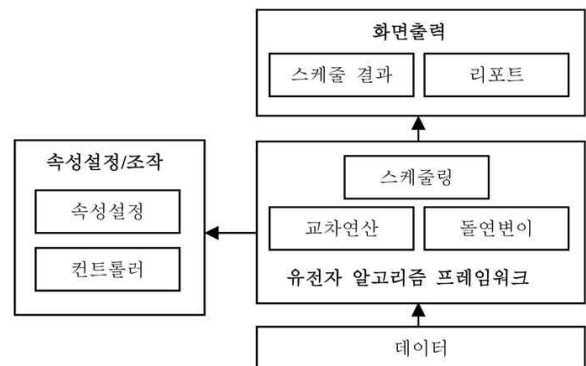
(그림 6) 적합도 평가

건에서 적합도를 평가하면 6,2,4,10,1,9,3,5,8,7은 415가 되어 엘리트1의 해로 보존되며, 이에 대한 경쟁 스케줄의 순열인 8,2,4,9,1,6,3,5,7,10로 적합도를 계산하면 382가 되어 E3로 선택하여 보존한다. 이러한 적합도 평가와 엘리트 보존 방식은 (그림 6)과 같다.

5. 시스템구현과 연구결과 분석

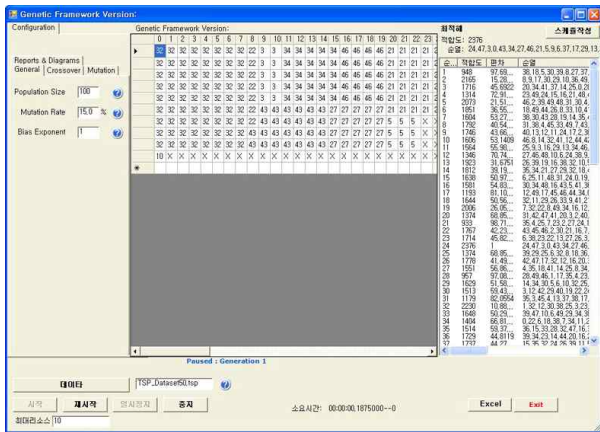
5.1 시스템 구현

본 논문에서는 Microsoft Visual studio 2008 VB.NET 환경에서 GAF(Genetic Algorithm Framework)를[21] 기반으로 스케줄링 시스템을 구현하였다. 구현된 스케줄링 시스템은 유전연산의 핵심이 되는 유전자 알고리즘 프레임워크, 유전 연산 속성의 설정/조작, 염색체 정보 데이터, 결과 출력 화면으로 구성되며 다음 (그림 7)은 구현된 스케줄링 시스템의 구성도이다.



(그림 7) 스케줄링 시스템 구성도

화면상의 좌측 영역은 유전연산의 속성을 변경할 수 있는 환경설정 영역이며, 가운데 영역은 최적 해에 대한 프로젝트 스케줄 테이블이다. 우측 영역은 결과 리포트 화면으로 프로젝트의 순열과 적합도 및 편차를 확인 할 수 있으며, 하단 영역은 유전연산의 시작과 종료를 할 수 있는 컨트롤이 위치한다. 다음 (그림 8)은 구현된 시스템 화면이다.



(그림 8) 화면 구성

(그림 9)의 화면은 스케줄링 결과 화면이다. 그림에서 세로축은 개발자 수, 가로축은 기간, 셀의 번호는 프로젝트의 번호를 나타낸다. 예를 들면 P32는 9명의 개발자로 9개월 동안 진행되며, P3는 5명의 개발자로 2개월 동안 진행되는 프로젝트이다.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
▶	32	32	32	32	32	32	32	32	22	3	3	34	34	34	34	34	46	46	46	46	21	21	21	21
	32	32	32	32	32	32	32	22	3	3	34	34	34	34	34	46	46	46	46	21	21	21	21	21
	32	32	32	32	32	32	32	22	3	3	34	34	34	34	34	46	46	46	46	21	21	21	21	21
	32	32	32	32	32	32	32	22	3	3	34	34	34	34	34	46	46	46	46	21	21	21	21	21
	32	32	32	32	32	32	32	22	43	43	43	43	43	43	43	27	27	27	27	27	21	21	21	21
	32	32	32	32	32	32	32	22	43	43	43	43	43	43	43	27	27	27	27	27	5	5	5	X
	32	32	32	32	32	32	32	43	43	43	43	43	43	43	43	27	27	27	27	27	5	5	5	X
	32	32	32	32	32	32	32	43	43	43	43	43	43	43	43	27	27	27	27	27	5	5	5	X

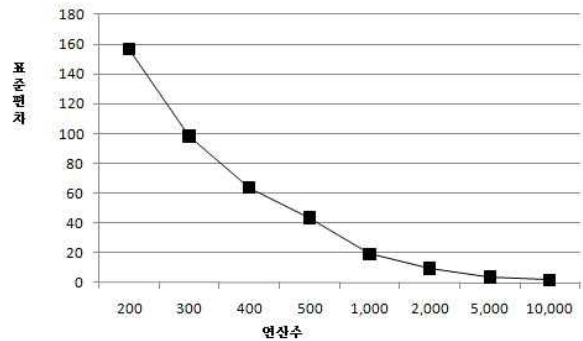
(그림 9) 스케줄링 결과

기존 연구들의 모형과 본 논문의 모형이 상이하여 성능의 직접적인 비교평가는 어려운바 본 논문에서는 E-기업의 최근 실제사례 해의 개선효과 위주로 성능 평가를 하고자 한다. 시스템의 성능 평가는 Microsoft Windows XP Professional 운영체제와 Intel Core Duo T7100 1.80GHz CPU의 환경에서 실행한다.

유전자 알고리즘의 성능을 보기 위하여 대상 프로젝트의 개수에 따른 연산시간의 변화와 연산횟수에 따른 적합도인 최대가치의 수렴 및 개선과정을 보고하고자 한다. <표 6>에서 연산횟수에 따른 각 해집단(population) 내 해들의 최대가치와 평균편차를 측정하는 것으로, 대상 프로젝트의 개수를 50

<표 6> 연산 반복에 따른 최대가치와 평균편차

No	연산횟수	최대가치	평균	평균편차
1	200	2,372	2,214	157.02
2	300	2,376	2,277	98.32
3	400	2,376	2,312	63.76
4	500	2,376	2,332	43.48
5	1,000	2,376	2,356	19.27
6	2,000	2,376	2,366	9.63
7	5,000	2,376	2,372	3.85
8	10,000	2,376	2,374	1.92



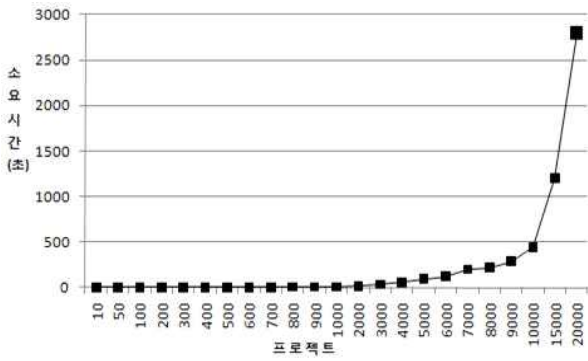
(그림 10) 연산 횟수에 따른 평균편차

개로 고정하고 유전연산의 반복횟수를 조정하며 최대가치, 평균, 평균편차를 보여주고 있다. 실행결과 유전 연산횟수가 증가함에 따라 최대가치와 가용한 개체군의 평균편차가 줄어들며 5,000회 이상에서 해집단이 수렴하며 평균편차도 적은 오차를 보여주고 있다. (그림 10)은 연산 횟수와 평균의 관계를 나타낸 그래프이다.

<표 7> 프로젝트 개수에 따른 유전연산 소요시간

No	프로젝트	소요시간(초)	증가율(%)
1	10	1.70	100
2	100	2.60	152
3	500	3.75	220
4	1,000	6.28	369
5	2,000	17.23	1,013
6	5,000	92.21	5,424
7	10,000	443.46	26,085
8	15,000	1200.46	70,615
9	20,000	2800.12	164,712

프로젝트 개수 변경에 따른 연산시간을 측정하기 위한 것으로, 유전연산 프로젝트의 개수를 조절하여 진행하고, 적합도가 1,000회 이상 변경이 없으면 연산 수행을 종료한다. <표 7>은 프로젝트 개수에 따라 알고리즘의 수행시간이 변화를 보여주고 있다. 약 15,000여 프로젝트에서 수행시간의 급속한 증가 현상을 볼 수 있으나 1,200초에 해당하는 짧은 시간에 해결이 가능한 것이다. (그림 11)은 프로젝트 개수와 유전연산에 소요된 시간의 관계를 나타낸 그래프이다.

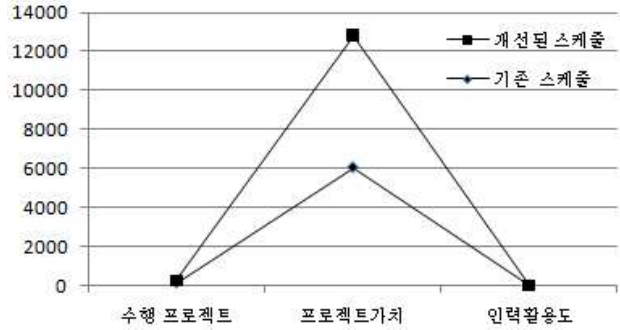


(그림 11) 프로젝트 개수에 따른 유전연산 소요시간

5.2 스케줄링 결과 검증

스케줄링 결과 검증을 위하여 오픈마켓 업체인 E사에서 2010년도 1월부터 8월까지의 월별 프로젝트 정보를 대상으로 기존 스케줄 결과와 본 논문의 시스템을 사용한 개선된 스케줄 결과를 비교하였다. <표 8>은 이에 대한 비교 결과 표이다.

1~8월까지의 기 수행된 스케줄과 시스템을 통하여 생성된 스케줄을 비교 결과 프로젝트 가치의 합이 739 증가하고, 인력활용도는 8% 증가하였다. (그림 12)는 기존 스케줄과 개선된 스케줄의 결과를 그래프로 나타낸다.



(그림 12) 스케줄 결과 비교

6. 결론

최근 오픈마켓 시장에서는 경쟁력 강화를 위하여 다양한 프로젝트가 진행되고 있으며 이에 대한 우선순위를 결정하기 위한 포트폴리오 관리가 중요시되고 있다. 그렇지만 포트폴리오 관리에는 많은 불확실한 요소가 내포되어 있어, 전문가에 의한 경험적 판단이나 프로젝트를 요청하는 팀의 요구 강도에 의하여 프로젝트의 우선순위가 결정되고 있다. 이러한 문제점으로 인해 비교적 가치가 작은 프로젝트가 우선적으로 진행되어 자원 낭비와 수익이 감소하게 되며, 결과적으로 시장 경쟁력 약화로 이어지게 된다. 기존의 포트폴리오 선정 방식은 주로 자원의 활용의 최대화를 목적으로 진행되며, 프로젝트 가치와 구현가능성에 대해서는 반영되지 않고, 유사프로젝트의 연속 진행 시 수행비용이 감소됨을 고려하지 못하고 있다.

본 논문에서는 오픈마켓의 효율적인 포트폴리오 관리를 위하여 포트폴리오 점수의 최대화를 목적으로 프로젝트 평가표를 활용하여 프로젝트 가치와 성공가능성을 측정하여 두 요소를 최대화시키는 방법과 프로젝트의 유사도를 결정하여 성공가능성을 보정하는 포트폴리오 관리모형을 제시하

<표 8> 기존 스케줄과 개선된 스케줄의 비교

차수	프로젝트	수행 PJT		가치		인력활용	
		기존	개선	기존	개선	기존	개선
1월	53	16	18	768	856	84%	95%
2월	56	15	18	720	868	86%	93%
3월	52	14	17	672	809	85%	92%
4월	53	15	17	727	815	86%	91%
5월	59	15	16	720	778	88%	94%
6월	54	16	18	768	867	89%	95%
7월	58	17	18	816	871	84%	93%
8월	61	18	19	852	918	83%	96%
합계		126	141	6043	6782	86%	94%

고, 유전자알고리즘을 이용한 스케줄링 시스템을 구현하여 오픈마켓 기업인 E사에 적용하였다. E사에 적용된 스케줄링 시스템을 검증을 위하여 성능평가를 실시하고, 기존 스케줄링 결과와 비교 분석하였다. 비교 분석 결과 프로젝트가치와 구현가능성 그리고 인력활용도가 증가함을 확인하였다.

본 논문의 개선점으로는 포트폴리오 점수를 산정하기 위해서 사용한 우선순위 결정 요인에 대하여, 정확성을 높이기 위하여 각 요인이 세분화되어야 하고, 기준 별 가중치를 산정하는 연구가 보완되어야 한다. 이와 함께 본 논문에서 고려하지 못한 법적 규제 및 특수한 관계에 의한 우선적 결정 요인과 스케줄링 후 프로젝트 정보와 가용인력 변경에 대해서 처리가 가능한 스케줄링이 추가로 연구되어야 한다.

참 고 문 헌

[1] Project Management Institute, "A Guide to the Project Management Body of Knowledge", Third Edition. pp.5-17, 2004.

[2] Ringuest, J. L. , "Conditional Stochastic Dominance in R&D Portfolio Selection", IEEE Transaction on Engineering Management, Vol.47, pp.478-484, Nov., 2000.

[3] Ghasemzadeh, F., "Project Portfolio Selection through Decision Support", Decision Support System, Vol.29, pp.73-88, July, 2000.

[4] Stummer, C., "Interactive R&D Portfolio Selection Considering Multiple Objective, Project Interdependencies, and Time: A Three-Phase Approach", Management of Engineering and Technology, Vol.2, pp.423-428, 2001.

[5] J. Chen and R. G. Askin, "Project selection, scheduling and resource allocation with time dependent returns", European Journal of Operational Research, Vol.193, pp.23-34, Feb., 2009.

[6] Hironori Kasahara, Seinosujke Narita, "Practical Multiprocessor Scheduling Algorithms for Efficient Parallel Processing", IEEE Transactions on computers, Vol.33, No.11, pp.1023-1029, Nov., 1984.

[7] Gonçalves, J.F, Mendes, J.J.M, Resende, M.G.C, "A genetic algorithm for the resource constrained multi-project scheduling problem", European Journal of Operational Research, Vol.189, No.3, pp.1171-1190, Sep., 2008.

[8] Paul R. Harper, Valter de Senna, Israel T. Vieira, Arjan K. Shahani, "A genetic algorithm for the project assignment problem", Computer & operations research, Vol.32, No.5, pp.1255-1265, May, 2005.

[9] Enrique Alba, J. Francisco Chicanoa, "Software project management with GAs", Elsevier in Information Sciences, Vol.177, No.11, pp.2380-2401, June, 2007.

[10] M Shepperd, C Schofield, "Estimating software project effort

using analogies", IEEE Transaction on Software Engineering, Vol.23, pp.736-743, Nov., 1997.

[11] Sadahiro Isoda, "Experiences of a Software Reuse Project", Journal of Systems and Software, Vol.30, No.3, pp. 71-186, Sep., 1995.

[12] M Petr Musilek, Witold Pedrycz, Giancarlo Succi, Marek Reformat, "Software cost estimation with fuzzy models", ACM SIGAPP Applied Computing Review, Vol.8, pp.24-29, Fall, 2000.

[13] Ali Idri, Alain Abran, "A Fuzzy Logic Based Set of Measures for Software Project Similarity: Validation and Possible Improvements", IEEE Transaction on Software Engineering, pp.85, Apr., 2001.

[14] Gada Kadoda, Michelle Cartwright, Liguang Chen, Martin Shepperd "Experiences Using Case-Based Reasoning to Predict Software Project Effort", Empirical Software Engineering Research Group, Jan., 2000.

[15] Rossella Bisio, Fabio Malabocchia, "Cost estimation of software projects through case base reasoning", Springer Berlin / Heidelberg, Vol.1010, pp.11-22, 1995.

[16] Robert W. Schwanke, "An intelligent tool for re-engineering software modularity", IEEE Computer Society Press, pp.83-92, May, 1991.

[17] Yong-hee Lee, Gun-Ho Lee "Development Process Improvement of Open-market System by using Agile Methodology", Korea Information Processing Society Vol.17D, No.3, pp.203-208, Aug., 2010.

[18] Capers Jones, "Reusability in Programming: A Survey of the State of the Art", IEEE Transactions on Software Engineering, Vol.SE-10, No.5, pp.488-493, 1984.

[19] Barry Boehm, "COCOMO II Model Definition Manual", The University of Southern California, 1997.

[20] Oliver, D. Smith, and J. Holland, "A study of permutation crossover operators on the traveling salesman problem", International Conference on Genetic Algorithms, pp.224-230, 1987.

[21] Rubicite Interactive Co., "Genetic Algorithm Framework", 2008. <http://www.rubicite.com/genetic/tutorial/>



이 용 희

e-mail : ta512k@etri.re.kr

2011년 숭실대학교 산업정보시스템공학 (박사)

2010년~현 재 한국전자통신연구원 연구원

2010년~현 재 ㈜에코시안 IT사업부 팀장

관심분야: 프로세스개선, 프로젝트관리,

개발방법론



이 건 호

e-mail : ghlee@ssu.ac.kr

1996년 U. of Iowa, Dept of Industrial Eng.

1997년~2004년 숭실대학교 산업공학과
전임강사

1999년~2004년 숭실대학교 산업정보시스
템공학과 조교수

2004년~2011년 숭실대학교 산업정보시스템공학과 부교수

2011년~현 재 숭실대학교 산업정보시스템공학과 교수

관심분야: 소프트웨어공학, 데이터마이닝, 기계학습