

<https://doi.org/10.7236/JIIBC.2021.21.5.189>

JIIBC 2021-5-25

## 대출 문제에서의 최선의 선택

### Best Choice in Loans Problem

이상운\*

Sang-Un Lee\*

**요약** 본 논문은 금리가 다른 다중 은행의 다중 국가에서 어느 은행에서 얼마의 대출을 받아야 년 상환액을 최소화할 수 있는지를 결정하는 대출선택 문제(CLP)를 다룬다. 이 문제에 대해 지금까지는 선형계획법과 같은 수학적 소프트웨어 패키지의 도움 없이는 해를 얻기가 사실상 불가능하였다. 본 논문에서는 수송문제에서 적용하고 있는 최소 금리부터 대출금을 결정하는 최소비용선택법(LCM)으로 초기 실현가능 해를 구하고, 교환 최적화 기법을 적용하여 초기 해를 개선하여 최적 해를 구하는 규칙을 가진 알고리즘을 CLP에 적용하였다. 제안된 알고리즘을 실험한 결과, LCM으로 얻은 초기 해에 대해 단지 최대 2회의 교환 최적화만으로도 최적 해를 얻을 수 있음을 보였다.

**Abstract** This paper discusses choice of loans problem(CLP) that is to minimize annual payment from which bank's borrows in multi-banks multi-nations with distinct interests. For the CLP, there is impossible to obtain the optimal solution actually without the help of mathematical software package as linear programming(LP). This paper applies the method used in transportation problem(TP) that finds initial feasible solution with selects minimum interest first, least cost method(LCM), to CLP. Result of experiment, the proposed algorithm can be obtains the optimal solution with at most two exchange optimization for LCM's initial feasible solution.

**Key Words** : choice of loans problem(CLP), linear programming(LP), least cost method(LCM), exchange optimization method(EOM)

### 1. 서론

다국적 판매망을 가진 회사의 사장이 다수의 국가에 신규 점포를 개설하여 판매망을 확충하려고 한다. 이때 신규 점포를 개설하는데 소요되는 비용은 전액 은행대출로 충당하고자 한다. 그러나 각 국가마다 각 점포의 위치와 연관된 위험요소 평가결과에 따라 각 은행은 상이한 이자율(금리)을 책정하고 있다. 이러한 상황에서 회사 사

장은 어느 은행에서 얼마의 대출을 하면 년 상환액을 최소화할 수 있는지를 결정하는 문제를 대출 선정문제(choice of loans problem, CLP)라 한다.<sup>[1,2]</sup>

지금까지는 CLP에 대해 전문가만이 활용할 수 있는 선형계획법(linear programming, LP)과 같은 수학적 프로그램 패키지 도움 없이는 해를 얻기가 쉽지 않았다. 또한 LP는 해를 어떻게 얻었는지, 해를 찾아가는 규칙을 제시하지 않고 단지 결과만을 제시하는 관계로 일반인들

\*정회원, 강릉원주대학교 과학기술대학 멀티미디어공학과  
접수일자 2021년 4월 2일, 수정완료 2021년 9월 4일  
게재확정일자 2021년 10월 8일

Received: 2 April, 2021 / Revised: 4 September, 2021 /  
Accepted: 8 October, 2021

\*Corresponding Author: [sulee@gwnu.ac.kr](mailto:sulee@gwnu.ac.kr)

Dept. of Multimedia Eng., Gangneung-Wonju National University, Korea

은 활용성이 떨어진다.

따라서 이러한 문제를 해결하기 위해 본 논문에서는 최근 대부분의 가정이나 회사에서 기본적으로 보유한 개인용 컴퓨터(PC)에 기본적으로 설치된 MS-Excel을 활용하여 CLP에 대해 빠르고, 정확하며, 손쉽게 최적 해를 찾아가는 규칙을 가진 알고리즘을 제안하고자 한다.

2장에서는 은행대출문제에 대한 문제 정의와 더불어 LP와 CPLEX로 해를 얻는 방법을 고찰해 본다. 3장에서는 MS-Excel을 활용하여 빠르고, 정확하며, 손쉽게 최적 해를 찾아가는 규칙을 가진 알고리즘을 제안하고, 4장에서는 제안된 방법을 적용하여 일반인이 자신의 대출계획을 스스로 결정할 수 있는지를 평가하여 본다.

## II. 은행대출문제 정의 및 관련연구

의류판매유통업의 사장은 런던, 뮌헨과 로마의 3곳에 신규 점포를 개설하고자 한다. 각 점포를 개설하는데 소요되는 비용은 각각 €2,500,000, €1,000,000과 €1,700,000이 소요되며, 이 소요비용은 3곳의 다른 은행으로부터 대출을 받아 충당하고자 한다. 각 점포의 위치와 연관된 위험요소 평가결과에 따라 각 은행은 표 1과 같이 상이한 이자율로 최대 €3,000,000을 8년간 대출할 수 있다고 결정하였다. 이 의류회사 사장은 어느 은행으로부터 대출을 받아야 년 상환액을 최소로 할 수 있는가?

Interest ( $i_{ij}$ )	Shops			Available loans ( $a_i$ )
	London	Munich	Rome	
Open costs( $o_j$ )	€ 2,500,000	€ 1,000,000	€ 1,700,000	
Bank #1	0.050 (5.0%)	0.065 (6.5%)	0.061 (6.1%)	3000000
Bank #2	0.052 (5.2%)	0.062 (6.2%)	0.062 (6.2%)	3000000
Bank #3	0.055 (5.5%)	0.058 (5.8%)	0.065 (6.5%)	3000000

그림 1. 예제 데이터  
Fig. 1. Example data

매장(shop)을  $s_j$ , 점포 개설 비용(opening cost)을  $o_j$ , 대출금(loan)을  $l_j$ , 은행의 대출가능총액(available loans)을  $a_i$ , 대출이자(interest)를  $i_{ij}$ , 대출기간(duration)을  $d$ 라 하면, CLP는 선형계획법(linear programming, LP)으로 표현하면 식 (1)의 목적함수를 찾고자 한다.

$$\text{minimize } \sum_{i=1}^m \sum_{j=1}^n l_j \times \frac{i_{ij}}{1 - (1 + i_{ij})^{-d}} \quad (1)$$

$$\text{subject to } \forall_j \in s_j : \sum_{i=1}^m l_j = o_j$$

$$\forall_i \in a_i : \sum_{j=1}^n l_j \leq a_i$$

년 상환액(annual payment)  $p$ 는 식 (2)로 계산된다.

$$p = l_j \times \frac{i_{ij}}{1 - (1 + i_{ij})^{-d}} \quad (2)$$

$$\text{where } l_j = \sum_{t=1}^d \frac{p}{(1 + i_{ij})^t} = d \times \frac{1 - (1 + i_{ij})^{-d}}{i_{ij}}$$

Guéret et al.<sup>[1]</sup>은 선형계획법(linear programming, LP)을, Edvall<sup>[2]</sup>은 C 언어로 구현된 simplex method인 CPLEX(IBM ILOG CPLEX Optimization Studio)를 이용하여 해를 구하고자 하였으며, 결과는 그림 2에 제시하였으며, 년 상환액은 € 822,180.66이다.

대출이자 ( $i_{ij}$ )	Shops			Available loans ( $a_i$ )
	London	Munich	Rome	
Open costs( $o_j$ )	€ 2,500,000	€ 1,000,000	€ 1,700,000	
Bank #1	0.050	0.065	0.061	€ 3,000,000
Bank #2	0.052	0.062	0.062	€ 3,000,000
Bank #3	0.055	0.058	0.065	€ 3,000,000

대출금 ( $l_j$ )	Shops			Total
	London	Munich	Rome	
Open costs( $o_j$ )	€ 2,500,000	€ 1,000,000	€ 1,700,000	€ 5,200,000
Bank #1	€ 2,500,000		€ 500,000	€ 3,000,000
Bank #2			€ 1,200,000	€ 1,200,000
Bank #3		€ 1,000,000		€ 1,000,000

년 상환액 ( $p$ )	Shops			Total
	London	Munich	Rome	
Bank #1	€ 386,804.53	€ 0.00	€ 80,836.93	
Bank #2	€ 0.00	€ 0.00	€ 194,775.57	
Bank #3	€ 0.00	€ 159,763.62	€ 0.00	
Total	€ 386,804.53	€ 159,763.62	€ 275,612.50	€ 822,180.66

그림 2.  $P_1$ 에 대한 LP와 CPLEX 결과

Fig. 2. Result of LP and CPLEX for  $P_1$

$P_1$  문제는 이자( $i_{ij}$ ) 오름차순으로  $\min\{a_i, o_j\}$ 를 만족하도록 식 (3)의 대출액  $l_{ij}$ 를 배정하는 Soomro et al.<sup>[3]</sup>이 이용한 LCM(least cost method)으로 쉽게 해를 얻을 수 있다. LCM을 적용한 결과는 그림 3과 같다.

$$l_{ij} = \min\{a_i, o_j\} \quad (3)$$

순위	$(b_i, s_j) = i_{ij}$	$a_i$	$o_j$	$l_j = \min\{a_i, o_j\}$
1	(1,1)=0.50	3,000,000	2,500,000	2,500,000
2	(2,1)=0.52	3,000,000	0	-
3	(3,1)=0.55	3,000,000	0	-
4	(3,2)=0.58	3,000,000	1,000,000	1,000,000
5	(1,3)=0.61	500,000	1,700,000	500,000
6	(2,2)=0.62	3,000,000	0	-
7	(2,3)=0.62	3,000,000	1,200,000	1,200,000
8	(1,2)=0.65	0	0	-
9	(3,3)=0.65	2,000,000	0	-

그림 3. 예제 데이터  $P_1$ 에 대한 LCM 결과  
 Fig. 3. Result of LCM for  $P_1$  example data

그러나 실제로는 이와 같은 LCM으로만 배정할 경우 최적의 해를 얻지 못하는 경우가 대부분이다. 따라서 CLP는 LCM으로 얻은 초기 해에 대해 최적화 단계를 추가로 수행해야만 최적 해를 얻을 수 있다. 따라서 본 논문에서는 수학적 소프트웨어 패키지인 LP나 CPLEX를 활용하지 않고도 쉽게 해를 얻을 수 있는 방법을 3장에서 제안한다.

### III. LCM 초기 해의 교환 최적화 방법

본 장에서는 수송문제(transportation problem, TP)에서 널리 사용되고 있는 LCM으로 얻은 초기 실현 가능 해(initial feasible solution)를 최적화시켜 최적 해를 얻는 방법을 제안한다. 이 방법은 수송문제의 해를 찾아가는 정확한 규칙을 제시한 다수의 논문들에서 기본적으로 채택하고 있는 방법이다.<sup>[4-8]</sup> 따라서 본 논문에서는 수송문제의 해를 얻는 가장 간단한 방법을 CLP에도 적용할 수 있음을 보인다.

LCM은 최소이자를 맨 처음 선택(최선의 선택)하고  $l_{ij} = \min\{a_i, o_j\}$ 를 만족하는 이자 오름차순으로 선택하는 방법으로 맨 마지막에 선택되는 셀은 최악의 이자가 되는 단점이 있다. 따라서 맨 마지막에 선택된 최악의 이자에 대한 큰 보상을 하기 위해 이전에 선택된 이자들이 약간의 양보를 하는 교환 최적화 방법을 적용하면 년 상환액을 최소로 하는 대출 은행의 최적 해를 얻을 수 있다.<sup>[4-8]</sup> 이는 그림 4와 같이 표현된다.

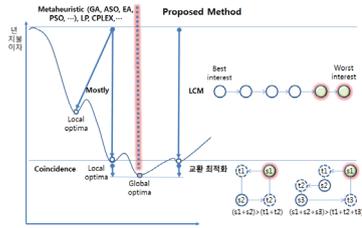


그림 4. LCM 초기 해의 교환 최적화 방법  
 Fig. 4. Exchange optimization method(EOM) for initial solution of LCM

제안된 알고리즘을 교환최적화 방법(exchange optimization method, EOM)이라 하며, 다음과 같이 수행된다.

Step 1. LCM으로 식 (3)의  $l_{ij} = \min\{a_i, o_j\}$ 를 배정한다.

여기서  $a_i, o_j$ 는 배정하고 남은 잔여량이다.

$l_{ij} > 0$ 셀들이 LCM 초기 해이다.

Step 2. Step 1에서 얻은  $l_{ij} > 0$ 에 대한  $i_{ij}$ 들 중 최대이자  $\max i_{ij}$ (맨 마지막에 배정된 셀의 이자)부터 내림차순으로 해당 셀을  $s_1$ , 동일 행에 대해  $i_{ij} < \max i_{ij}$ 인 셀을  $t_1$ ,  $t_1$  셀의 열에서 선택된 셀을  $s_2$ ,  $s_2$  셀의 행,  $s_1$  셀의 열을  $t_2$ 로 설정

if  $(l_{s_1} + l_{s_2}) > (l_{t_1} + l_{t_2})$  then  $\delta = \min\{l_{s_1}, l_{s_2}\}$

$s_1 : l_{s_1} - \delta, t_1 : l_{t_1} + \delta, s_2 : l_{s_2} - \delta, t_2 : l_{t_2} + \delta$

/\* 2-opt \*/

if  $(l_{s_1} + l_{s_2} + l_{s_3}) > (l_{t_1} + l_{t_2} + l_{t_3})$  then

$\delta = \min\{l_{s_1}, l_{s_2}, l_{s_3}\}$   $s_1 : l_{s_1} - \delta, t_1 : l_{t_1} + \delta,$

$s_2 : l_{s_2} - \delta, t_2 : l_{t_2} + \delta, s_3 : l_{s_3} - \delta, t_3 : l_{t_3} + \delta.$

/\* 3-opt \*/

그림 1의  $P_1$  문제의 대출이자( $i_{ij}$ )가 그림 5와 같이 변경된  $P_2$  문제에 제안된 EOM을 적용하여 보자.  $P_2$  문제에 EOM을 적용한 결과는 그림 6에 제시하였다.

대출이자 ( $i_{ij}$ )	Shops			Available loans ( $a_i$ )
	London	Munich	Rome	
Open costs( $o_j$ )	€ 2,500,000	€ 1,000,000	1,700,000	€
Bank #1	0.050	0.057	0.051	€ 3,000,000
Bank #2	0.052	0.055	0.054	€ 3,000,000
Bank #3	0.054	0.058	0.055	€ 3,000,000

그림 5.  $P_2$  문제  
 Fig. 5.  $P_2$  problem

Shop	London	Munich	Rome	Available loans ( $a_i$ )
Open costs( $o_j$ )	€ 2,500,000	€ 1,000,000	€ 1,700,000	
Bank #1	0.050	0.057	0.051	€ 3,000,000
Bank #2	0.052	0.055	0.054	€ 3,000,000
Bank #3	0.054	0.058	0.055	€ 3,000,000

Bank #1	€ 2,500,000	€ 0	€ 500,000	€ 3,000,000
Bank #2	€ 0	€ 0	€ 1,200,000	€ 1,200,000
Bank #3	€ 0	€ 1,000,000	€ 0	€ 1,000,000
Total	€ 2,500,000	€ 1,000,000	€ 1,700,000	€ 5,200,000

Bank #1	€ 386,804.53	€ 0.00	€ 77,673.93	
Bank #2	€ 0.00	€ 0.00	188679.8205	
Bank #3	€ 0.00	€ 159,763.62	€ 0.00	
Total	€ 386,804.53	€ 159,763.62	€ 266,353.75	€ 812,921.91

(a) LCM 초기 실현 가능 해

Bank #1	€ 1,300,000	€ 0.00	€ 1,700,000	(0.054+0.050)-(0.052+0.051)=0.001 min(1,200,000, 2,500,000)=1,200,000
Bank #2	€ 1,200,000	€ 0.00	€ 0	
Bank #3	€ 0.00	€ 1,000,000	€ 0.00	
Total	€ 2,500,000	€ 1,000,000	€ 1,700,000	

Bank #1	€ 201,138.36	€ 0.00	€ 264,091.36	
Bank #2	€ 187,170.13	€ 0.00	€ 0.00	
Bank #3	€ 0.00	€ 159,763.62	€ 0.00	
Total	€ 388,308.48	€ 159,763.62	€ 264,091.36	€ 812,163.47

(b) 1<sup>st</sup> 2-opt

Bank #1	€ 1,300,000	€ 0.00	€ 1,700,000	(0.058+0.052)-(0.055+0.054)=0.001 min(1,000,000, 1,200,000)=1,000,000
Bank #2	€ 200,000	€ 1,000,000	€ 0.00	
Bank #3	€ 1,000,000	€ 0.00	€ 0.00	
Total	€ 2,500,000	€ 1,000,000	€ 1,700,000	

Bank #1	€ 201,138.36	€ 0.00	€ 264,091.36	
Bank #2	€ 31,195.02	€ 157,864.01	€ 0.00	
Bank #3	€ 157,233.18	€ 0.00	€ 0.00	
Total	€ 389,566.56	€ 157,864.01	€ 264,091.36	€ 811,521.94

(c) 2<sup>nd</sup> 2-opt

그림 6.  $P_2$  문제에 대한 EOM 결과  
Fig. 6. Result of EOM for  $P_2$  problem

EOM으로 얻은 LCM 초기 실현 가능 해(년 상환액)는 € 812,921.91임을 알 수 있다. 그러나 첫 번째 2-opt 결과 € 812,163.47로 € 758.44를 감소시켰으며, 두 번째 2-opt 결과 다시 €641.53을 절감시켜 LCM 초기 해에 비해 총 € 1,399.97을 절감하였다.

#### IV. 적용 및 결과분석

본 장에서는 그림 7의 예제 데이터들에 대해 EOM을 실험하였으며, 실험 결과는 그림 8과 그림 9에 제시하였다. 실험 데이터는 Lee<sup>[5]</sup>에 제시된 문제를 변형시켜 생성하였다.

실험에 적용된 대출기간( $d$ )은 8년으로 확정된 값을 적용하였다. 본 장에서는 LP 또는 CPLEX와 같은 전문 가들만이 사용할 수 있는 수학적 소프트웨어 패키지를 활용하지 않고, 단순히 MS-Excel 만으로도 일반인이 다중원천 은행들로부터 년 상환액을 최소로 할 수 있는 대출액을 선택할 수 있는 범용성이 뛰어난 가장 쉬운 방법을 제안하였다.

CLP는 수기식으로는 해를 찾는 데 대부분 실패할 수 있다. 따라서 지금까지는 LP 또는 CPLEX 소프트웨어 패키지의 도움을 받아 선형방정식에 대한 수 많은 시행 결과 해를 얻을 수 있었다.

본 논문에서는 MS-Excel을 활용하여 수기식으로도 해를 빠르고 정확하며 간단히 찾아가는 명확한 규칙을 가진 알고리즘이 존재함을 보였다. 제안된 EOM은 기존의 LCM으로는 최적 해를 찾지 못하고 최적 해에 근사한 해를 찾는 단점을 보완하기 위해 교환 최적화 기법을 적용하였음을 알 수 있었으며, 이 결과 다중 은행들 중 최소의 년 상환액(최적 해)을 얻을 수 있도록 어느 은행에서 얼마를 대출받아야 하는지를 쉽게 결정할 수 있었다.

Shop	Shop #1	Shop #2	Shop #3	Shop #4	Shop #5	Shop #6	Available loans ( $a_i$ )
Open costs( $o_j$ )	€ 4,000,000	€ 3,500,000	€ 2,500,000	€ 2,000,000	€ 6,000,000	€ 3,000,000	
Bank #1	0.0328	0.0331	0.0344	0.0355	0.0338	0.0343	€ 6,000,000
Bank #2	0.0332	0.0356	0.0344	0.0334	0.0328	0.0350	€ 7,000,000
Bank #3	0.0344	0.0356	0.0324	0.0352	0.0336	0.0328	€ 8,000,000

(a)  $P_3$

Shop	Shop #1	Shop #2	Shop #3	Shop #4	Available loans ( $a_i$ )
Open costs( $o_j$ )	€ 1,000,000	€ 1,500,000	€ 1,200,000	€ 1,500,000	
Bank #1	0.0010	0.0030	0.0025	0.0015	€ 1,400,000
Bank #2	0.0020	0.0015	0.0020	0.0010	€ 1,000,000
Bank #3	0.0010	0.0030	0.0020	0.0020	€ 1,500,000
Bank #4	0.0030	0.0040	0.0035	0.0045	€ 1,300,000

(b)  $P_4$

그림 7. 실험 데이터  
Fig. 7. Experimental data

Shop	Shop #1	Shop #2	Shop #3	Shop #4	Shop #5	Shop #6	Available loans ( $a_i$ )
Open costs( $o_i$ )	€ 4,000,000	€ 3,500,000	€ 2,500,000	€ 2,000,000	€ 6,000,000	€ 3,000,000	
Bank #1	€ 4,000,000	€ 2,000,000					€ 0
Bank #2				€ 1,000,000	€ 6,000,000		€ 0
Bank #3		€ 1,500,000	€ 2,500,000	€ 1,000,000		€ 3,000,000	€ 0
총 대출액	€ 4,000,000	€ 3,500,000	€ 2,500,000	€ 2,000,000	€ 6,000,000	€ 3,000,000	€ 21,000,000

Bank #1	€ 576,575.62	€ 288,650.58	€ 0.00	€ 0.00	€ 0.00	€ 0.00	
Bank #2	€ 0.00	€ 0.00	€ 0.00	€ 144,506.79	€ 864,863.43	€ 0.00	
Bank #3	€ 0.00	€ 218,761.74	€ 359,755.59	€ 145,598.10	€ 0.00	€ 432,431.72	
년 상환액	€ 576,575.62	€ 507,412.32	€ 359,755.59	€ 290,104.89	€ 864,863.43	€ 432,431.72	€ 3,031,143.57

(a) LCM 초기 실현 가능 해

Bank #1	€ 4,000,000	€ 2,000,000					(0.0352+0.0328)-(0.0336+0.0334)=0.010
Bank #2				€ 2,000,000	€ 5,000,000		
Bank #3		€ 1,500,000	€ 2,500,000	€ 0	€ 1,000,000	€ 3,000,000	
총 대출액	€ 4,000,000	€ 3,500,000	€ 2,500,000	€ 2,000,000	€ 6,000,000	€ 3,000,000	min(1,000,000, 6,000,000)=1,000,000

Bank #1	€ 576,575.62	€ 288,650.58	€ 0.00	€ 0.00	€ 0.00	€ 0.00	
Bank #2	€ 0.00	€ 0.00	€ 0.00	€ 289,013.57	€ 720,719.53	€ 0.00	
Bank #3	€ 0.00	€ 218,761.74	€ 359,755.59	€ 0.00	€ 144,627.84	€ 432,431.72	
년 상환액	€ 576,575.62	€ 507,412.32	€ 359,755.59	€ 289,013.57	€ 865,347.37	€ 432,431.72	€ 3,030,536.19

(b) 1<sup>st</sup> 2-opt

Bank #1	€ 2,500,000	€ 3,500,000					(0.0356+0.0328+0.0328)-(0.0336+0.0332+0.0331)=0.0003
Bank #2	€ 1,500,000			€ 2,000,000	€ 3,500,000		
Bank #3		€ 0	€ 2,500,000	€ 0	€ 2,500,000	€ 3,000,000	
총 대출액	€ 4,000,000	€ 3,500,000	€ 2,500,000	€ 2,000,000	€ 6,000,000	€ 3,000,000	min(1,500,000, 5,000,000, 4,000,000)=1,500,000

Bank #1	€ 360,359.76	€ 505,138.51	€ 0.00	€ 0.00	€ 0.00	€ 0.00	
Bank #2	€ 216,578.66	€ 0.00	€ 0.00	€ 289,013.57	€ 504,503.67	€ 0.00	
Bank #3	€ 0.00	€ 0.00	€ 359,755.59	€ 0.00	€ 361,569.61	€ 432,431.72	
년 상환액	€ 576,938.43	€ 505,138.51	€ 359,755.59	€ 289,013.57	€ 866,073.28	€ 432,431.72	€ 3,029,351.10

(c) 2<sup>nd</sup> 3-opt

그림 8.  $P_3$  문제에 대한 EOM 결과

Fig. 8. Result of EOM for  $P_3$  problem

Shop	Shop #1	Shop #2	Shop #3	Shop #4	Available loans ( $a_i$ )
Open costs( $o_i$ )	€ 1,000,000	€ 1,500,000	€ 1,200,000	€ 1,500,000	
Bank #1		€ 200,000	€ 700,000	€ 500,000	€ 0
Bank #2				€ 1,000,000	€ 0
Bank #3	€ 1,000,000		€ 500,000		€ 0
Bank #4		€ 1,300,000			€ 0
총 대출액	€ 1,000,000	€ 1,500,000	€ 1,200,000	€ 1,500,000	€ 5,200,000

Bank #1	€ 0.00	€ 25,338.68	€ 88,487.24	€ 62,922.61	
Bank #2	€ 0.00	€ 0.00	€ 0.00	€ 125,563.16	
Bank #3	€ 125,563.16	€ 0.00	€ 63,063.81	€ 0.00	
Bank #4	€ 0.00	€ 165,438.62	€ 0.00	€ 0.00	
년 상환액	€ 125,563.16	€ 190,777.30	€ 151,551.05	€ 188,485.77	€ 656,377.28

(a) LCM 초기 실현 가능 해

Bank #1		€ 0	€ 700,000	€ 700,000	(0.0010+0.0030)-(0.0015+0.0015)=0.010
Bank #2		€ 200,000		€ 800,000	
Bank #3	€ 1,000,000		€ 500,000		
Bank #4		€ 1,300,000			min(200,000, 1,000,000)=200,000
총 대출액	€ 1,000,000	€ 1,500,000	€ 1,200,000	€ 1,500,000	

Bank #1	€ 0.00	€ 0.00	€ 88,487.24	€ 88,091.66	
Bank #2	€ 0.00	€ 25,169.05	€ 0.00	€ 100,450.52	
Bank #3	€ 125,563.16	€ 0.00	€ 63,063.81	€ 0.00	
Bank #4	€ 0.00	€ 165,438.62	€ 0.00	€ 0.00	
년 상환액	€ 125,563.16	€ 190,607.67	€ 151,551.05	€ 188,542.18	€ 656,264.06

(b) 1<sup>st</sup> 2-opt

Bank #1	€ 700,000	€ 0	€ 0	€ 700,000	$(0.0025+0.0010)-(0.0020+0.0010)=0.0005$ $\min(700,000, 1,000,000)=700,000$
Bank #2		€ 200,000		€ 800,000	
Bank #3	€ 300,000		€ 1,200,000		
Bank #4		€ 1,300,000			
총 대출액	€ 1,000,000	€ 1,500,000	€ 1,200,000	€ 1,500,000	
Bank #1	€ 87,894.21	€ 0.00	€ 0.00	€ 88,091.66	$€ 656,066.15$
Bank #2	€ 0.00	€ 25,169.05	€ 0.00	€ 100,450.52	
Bank #3	€ 37,668.95	€ 0.00	€ 151,353.15	€ 0.00	
Bank #4	€ 0.00	€ 165,438.62	€ 0.00	€ 0.00	
년 상환액	€ 125,563.16	€ 190,607.67	€ 151,353.15	€ 188,542.18	

(c) 2<sup>nd</sup> 2-opt

그림 9.  $P_4$  문제에 대한 EOM 결과

Fig. 9. Result of EOM for  $P_4$  problem

## V. 결 론

본 논문에서는 대출이자(금리)가 서로 다른 다중 은행들로부터 대출을 받고자 하는 경우 년 상환액을 최소화할 수 있도록 어느 은행으로부터 얼마를 대출받아야 하는지를 결정하는 CLP에 대해 빠르고 정확하면서도 단순한 규칙을 가지고 최적 해를 찾아가는 방법을 제안하였다.

지금까지는 CLP에 대해서는 LP 또는 CPLEX와 같은 수학적 소프트웨어 패키지의 도움 없이는 최적 해를 얻기가 사실상 불가능하였다. 또한 매년 전문가의 도움을 받기도 경제적손실로 인해 쉽지 않다. 따라서 LP 또는 CPLEX를 활용하지 못하는 일반인들도 손쉽게 자신의 은행 대출을 최적으로 계획하여 년 상환액을 최소화할 수 있는 방법이 절실히 요구되고 있었다.

이러한 요구사항에 부응하기 위해, 본 논문에서는 MS-Excel을 활용하여 일반인들도 손쉽게 해답을 찾을 수 있도록 최소이자부터 오름차순으로 대출금을 결정하고, 맨 마지막에 결정된 최악의 경우(최대 이자)를 보다 낮추기 위해 교환 최적화 기법을 적용하여 최적 해를 찾는 간단한 방법을 제시하였다.

제안된 알고리즘을 실험한 결과, 교환 최적화는 “0”회 또는 “2”회 만으로도 최적 해를 찾을 수 있음을 보였다. 따라서 제안된 EOM을 은행대출에 있어서 범용성을 가진 알고리즘으로 활용하면 경제적인 이득을 최대로 얻을 수 있을 것이다.

## References

[1] C. Guéret, X. Prins, and M. Sevaux, "Applications of Optimization with Xpress-MP: 13.1 Choice of Loans,"

Dash Optimization Ltd., pp. 195-197, Feb. 2005.

- [2] M. Edvall, "Choice of Loan Portfolio," Tomlab Optimization Inc, [http://tomsym.com/examples/tomsym\\_choiceofloans.html](http://tomsym.com/examples/tomsym_choiceofloans.html), Apr. 2009.
- [3] A. S. Soomro, M. Junaid, and G. Tularam, "Modified Vogel's Approximation Method For Solving Transportation Problems," *Mathematical Theory and Modeling*, Vol. 5, No. 4, pp. 32-42, May 2015.
- [4] S. U. Lee and M. B. Choi, "Optimal Algorithm for Transshipment Problem," *The Journal of The Institute of Internet, Broadcasting and Communication(IIBC)*, Vol. 13, No. 1, pp. 153-162, Feb. 2013. <https://doi.org/10.7236/JIIBC.2013.13.1.153>
- [5] S. U. Lee, "Optimal Solution for Transportation Problems," *The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)*, Vol. 13, No. 2, pp. 93-102, Apr. 2013. <https://doi.org/10.7236/JIIBC.2013.13.2.93>
- [6] S. U. Lee and M. B. Choi, "Simple Solution for Multi-commodity Transportation Problem," *The Journal of The Institute of Internet, Broadcasting and Communication(IIBC)*, Vol. 13, No. 5, pp. 173-181, Oct. 2013. <https://doi.org/10.7236/JIIBC.2013.13.5.173>
- [7] S. U. Lee, "The Assignment-Swap Algorithm for Large-scale Transportation Problem with Incomplete Cost Lists," *Journal of The Korea Society of Computer and Information(KSCSI)*, Vol. 20, No. 6, pp. 51-58, Jun. 2015. <https://doi.org/10.9708/jksoci.2015.20.6.051>
- [8] S. U. Lee, "Simple Algorithm for Large-scale Unbalanced Transportation Problem," *The Journal of The Institute of Internet, Broadcasting and Communication(IIBC)*, Vol. 15, No. 4, pp. 223- 230, Aug. 2015. <https://doi.org/10.7236/JIIBC.2015.15.4.223>

## 저 자 소 개

### 이 상 운(정회원)



- 1987년 : 한국항공대학교 항공전자공학과 (학사)
- 1997년 : 경상대학교 컴퓨터과학과 (석사)
- 2001년 : 경상대학교 컴퓨터과학과 (박사)
- 2003년 : 강원도립대학 컴퓨터응용과

전임강사

- 2004년 ~ 2007년 2월 : 국립원주대학 여성교양과 조교수
- 2007년 3월 ~ 2015년 3월 : 강릉원주대학교 멀티미디어공학과 부교수
- 2015년 4월 ~ 현재 : 강릉원주대학교 멀티미디어공학과 정교수
- 관심분야 : 소프트웨어 프로젝트 관리, 개발 방법론, 분석과 설계 방법론, 시험 및 품질보증, 소프트웨어 신뢰성, 인공지능과 빅데이터 분석, 최적화 알고리즘
- e-mail : [sulee@gwnu.ac.kr](mailto:sulee@gwnu.ac.kr)