Journal of The Korea Society of Computer and Information
Vol. 20 No. 12, pp. 75-82, December 2015
www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2015.20.12.075

One-Sided Optimal Assignment and Swap Algorithm for Two-Sided Optimization of Assignment Problem

Sang-Un Lee*

Abstract

Generally, the optimal solution of assignment problem can be obtained by Hungarian algorithm of two-sided optimization with time complexity $O(n^4)$. This paper suggests one-sided optimal assignment and swap optimization algorithm with time complexity $O(n^2)$ can be achieve the goal of two-sided optimization. This algorithm selects the minimum cost for each row, and reassigns over-assigned to under-assigned cell. Next, that verifies the existence of swap optimization candidates, and swap optimizes with $k-opt(k=2,3)$. For 27 experimental data, the swap-optimization performs only 22% of data, and 78% of data can be get the two-sided optimal result through one-sided optimal result. Also, that can be improves on the solution of best known solution for partial problems.

▸ Keywords : Hungarian Algorithm, Minimum Cost, One-sided Optimization, Reassignment, Swap

• First Author: Sang-Un Lee
*Sang-Un Lee (sulee@gwnu.ac.kr), Dept. of Multimedia Engineering, Gangneung-Wonju National University
• Received: 2015. 08. 31, Revised: 2015. 09. 21, Accepted: 2015. 09. 29.

I. Introduction

다수의 작업들 (Jobs, $J_i, i=1,2,\cdots,m$)과 기계 (Machines, $M_j, j=1,2,\cdots,n$)가 존재하며, 각 작업을 특정 기계에서 수행하는 작업시간 (또는 비용, c_{ij})이 다르며, 하나의 작업은 반드시 하나의 기계에서만 작업이 이루어진다. 이 경우, 최소 처리시간 (또는 비용) 합인 최적 해 $_{\min}z=\sum_{i=1}^{m}\sum_{j=1}^{n}c_{ij}x_{ij}$, $x_{ij}=1$을 찾도록 각 작업을 기계들에 배정하는 문제를 할당 문제 (assignment problem)라 한다[1-3].

할당 문제는 대부분 헝가리안 (Hungarian) 알고리즘[2,3]을 적용하고 있으며, 일부는 유전자 알고리즘[4]을 시도하는 경우도 있다. 헝가리안 알고리즘은 Kuhn이 1955년도에 제안하였으며, 헝가리 수학자 König와 Egerváry의 연구에 전적으로 기반하고 있어 헝가리안 알고리즘이라 칭한다. 이 알고리즘을 Kuhn-Munkres 또는 Munkres 할당 알고리즘이라고도 한다[2].

헝가리안 알고리즘은 양측-최적화 방법을 수행 복잡도 $O(n^4)$으로 수행하며, 균형 할당 문제에 대해서는 최적 해를 항상 찾을 수 있다고 알려져 있다[1]. 그러나 불균형 할당 문제에 대해서는 최적 해를 찾지 못할 가능성이 있기 때문에 비용이 모두 0인 가상의 행이나 열을 추가하여 균형 할당문제로 변환시킨다[3]. Edmonds와 Krap은 헝가리안 알고리즘의 수행 복잡도를 $O(n^3)$으로 향상시킨 알고리즘을 제안하였다[2].

할당문제와 관련하여 지금까지 제안된 알고리즘들은 $_{\min}z$에만 주안점을 두고 있으며, 모든 작업이 끝나는 작업종료시간 (makespan, s)을 최소화시키는 평가기준은 고려하지 않는다. 그러나 n개의 작업이 모두 종료되는 작업종료시간을 최소로 하는 순환시간 (cycle time) 최소화로 작업의 생산율을 높일 수 있는 중요한 평가요소로 이 기준도 함께 고려해야만 한다.

본 논문은 할당문제의 최소 비용 합 $_{\min}z$ 뿐 아니라 최소의 작업종료시간 $_{\min}s$을 수행 복잡도 $O(n^2)$으로 찾을 수 있는 알고리즘을 제안한다. 2장에서는 헝가리안 알고리즘을 살펴보고 문제점을 고찰해 본다. 3장에서는 단측-최적화와 교환 (swap) 방법을 적용하여 할당 문제의 양측-최적화 목표를 달성하는 알고리즘을 제안한다. 4장에서는 다양한 할당 문제 사례들에 제안된 알고리즘을 적용하여 최적 해를 찾는지 평가해 본다.

II. Related Works and Problems

할당 문제는 하나의 작업은비용을 최소로 하는 하나의 기계만을 선택하여야만 한다. 이는 기계에 일을 부여하는 경우, 사람에게 임무를 부여하는 경우 등에 일반적으로 적용된다. 할당 문제는 식 (1)의 조건을 만족하는 최적 해를 찾는다[3].

$$
\begin{aligned}
z &= {}_{\min}\sum_{i=1}^{m}\sum_{j=1}^{n}c_{ij}x_{ij} \qquad (1)\\
\text{s.t. } & \sum_{i=1}^{m}x_{ij}=1 \text{ for } i=1,2,\cdots,m.\\
& \sum_{j=1}^{n}x_{ij}=1 \text{ for } j=1,2,\cdots,n.\\
& x_{ij}\geq 0, \text{ for } \forall_{ij}.
\end{aligned}
$$

할당 문제의 최적 해를 찾는 헝가리안 알고리즘은 그림 1과 같이 수행된다.

$m\times n(m=n)$ 비용 행렬
Step 1./* 수행 복잡도 : $O(n^2)$
    for $i=1$ to m
        최소 비용 $_{\min}c_i$ 선택.
        for $j=1$ to n
            $c_{ij}=c_{ij}-{}_{\min}c_i$
        end
    end
    for $j=1$ to n
        최소 비용 $_{\min}c_j$ 선택
        $i=1$ to m
            $c_{ij}=c_{ij}-{}_{\min}c_j$
        end
    end
Step 2./* 수행 복잡도 : $O(n^2)$
    모든 $c_{ij}=0$을 포함하는 최소한의 선을 그음.
    /* 선의 개수 : $|l|$
    if $|l|=m$then go to Step 4.
    else if $|l|<m$then go to Step 3.
Step 3./* 수행 복잡도 : $O(n^3)$
    (1) 선에 포함되지 않은 $c_{ij}>0$들로 구성된 감소된 비용 행렬에서 최소 비용 $_{\min}c_{ij}$ 선택, $c_{ij}>0$에 대해 $c_{ij}=c_{ij}-{}_{\min}c_{ij}$ 계산.
    (2) 행과 열의 선이 교차된 $c_{ij}>0$에 대해 $c_{ij}=c_{ij}+c_{ij}$ 계산.
    go to Step 2.
Step 4.각 열에서 0이 중첩되지 않게 1개씩 선택, 선택된 셀의 비용을 모두 합하여 최적해 z를 얻음.

Fig. 1. Hungarian Algorithm

그림 2는 Kumar[5]에서 인용된 할당 문제이다. 행은 작업을, 열은 기계를, 행렬의 값 은 작업 수행비용이다. 이 문제는 4개의 작업을 4개의 기계에 중복되지 않게 할당하여 총 작업비용을 최소화시키는 제약조건을 만족하는 최적 해를 찾아야 한다.

c_{ij}		Machines			
		1	2	3	4
Jobs	1	1	4	6	3
	2	8	7	10	9
	3	4	5	11	7
	4	6	7	8	5

Fig. 2. A_1 Assignment Problem

Kumar[5]는 그림 2의 A_1 할당문제에 대해 헝가리안 알고리즘을 적용하여 최적 해를 찾은 결과는 그림 3과 같다. Step 2를 2회, Step 3을 1회 수행하여 0을 모두 포함하는 최소 선을 m개 얻었으며, 작업을 $x_{11}=1, x_{23}=1, x_{32}=1, x_{44}=1$로 할당하여 Step 4에서 최적해 $z={}_{\min}\sum_{i=1}^{m}\sum_{j=1}^{n}c_{ij}x_{ij}=1+10+5+5=21$, $s={}_{\max}\{1,10,5,5\}=10$ 을 얻었다.

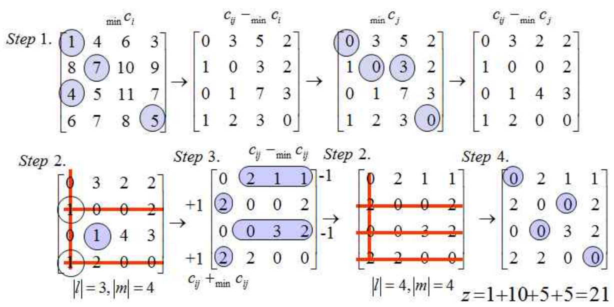


Fig. 3. Hungarian Algorithm for A_1

헝가리안 알고리즘은 최종적으로 얻은 직선들에 대해 행과 열을 1:1로 재배정해야 하는 어려움이 있으며, $_{\min}s$는 고려하지 않고, 단지 $_{\min}z$를 찾기 위해 선의 개수가 n개가 될 때까지 수행 복잡도 $O(n^4)$으로 양측-최적화를 수행한다[6]. 또한, 선을 긋는 규칙이 애매모호하게 정의하고 있어 Singh[7]은 보다 쉽고 명확한 규칙으로 선을 긋는 방법을 제안하기도 하였다. 만약, 동일한 $_{\min}z$를 가지면서 $_{\min}s$를 갖는 별개의 해가 존재한다면 이 별개의 해를 선택하는 것이 보다 최선의 방법이 될 수 있다. Lee[8]는 헝가리안 알고리즘의 복잡도를 개선하는 한 가지 방법으로 행에 대해 단측-최적화를 수행하고, 동일 열에 중복되는 경우 최소 증가분을 가진 값이 이동하는 방법을 제안하였다.

III. One-Sided Optimal Assignment and Swap Algorithm

본 장에서는 헝가리안 알고리즘의 수행 복잡도 $O(n^4)$을 $O(n^2)$으로 향상시키면서도 $_{\min}z$와 $_{\min}s$를 얻는 알고리즘을 제안한다. 여기서는 중복 선택된 경우 어느 것을 이동시킬 것인가에 대해 Lee[8]와는 다른 접근법을 제안한다.

헝가리안 알고리즘은 양측-최적화를 수행하는 반면에, 제안된 알고리즘은 행에 대해서만 배정하는 단측-최적화를 수행하고, 교환 최적화 과정을 다시 수행하는 2단계를 수행하는 전략으로, 단측-최적화 배정과 교환 (one-sided optimal assignment and swap, OOAS) 알고리즘이라 하자. OOAS 알고리즘은 다음과 같이 수행된다.

Phase I. One-sided Optimal Assignment

for $i=1$ to n /* i행: 수행 복잡도 $O(n^2)$ */
  $_{\min}c_{ij}$ 선택, $M \leftarrow c_{ij}$
end
$c_{ij} \in M, c'_{ij} \notin M$
for $j=1$ to n /* j열: 수행 복잡도 $O(n^2)$ */
  if $|c_{ij}| = \{\phi\}$ then skip
  else if $|c_{ij}|=1$ then $c'_{ij} > c_{ij}$인 c'_{ij} 셀 삭제
  else if $2 \le |c_{ij}| < n$ then $\forall c'_{ij}$ 셀 삭제
  else if $|c_{ij}|=n$ then skip
end
c_{ij}^0: $c_{ij} \in M$, c_{ij}^+: next cost $c'_{ij} \notin M$
until $\forall_j, |c_{ij}|=1$ do
  $2 \le |c_{ij}| = k \le n$인 j열에 대해, $\forall_j, c'_{ij}$ 셀 삭제
  $_{\min}\{c_{ij}^+ - c_{ij}^0\}$ 인 $k-1$개를 c_{ij}^+ 셀로 이동, c_{ij}^0 셀 삭제
end

Phase II. Swap Optimization

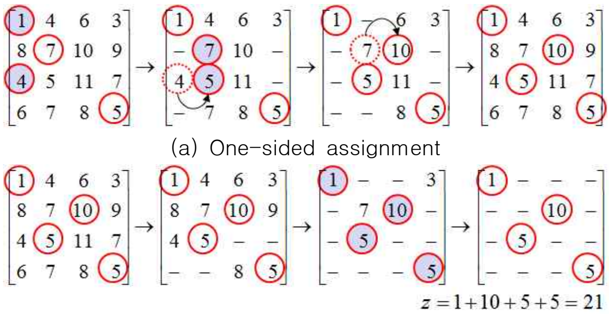
Step 1. Swap Preparation (Matrix Reduction)
  $i=1,2$ $j=1,2$에 대해 $c_{11} \in M$, $c_{12} \notin M$, $c_{21} \notin M$, $c_{22} \in M$이라 하자.
  for $i=1$ to n
    if $c_{12} > {}_{\max}\{c_{11}, c22\}$ then c_{12} 셀 삭제
    else If $c_{21} > {}_{\max}\{c_{11}, c22\}$ then c_{21} 셀 삭제
    else if $(c_{12}+c_{21}) > (c_{11}+c_{22})$ then c_{12}, c_{21} 셀 삭제
    else if $(c_{12}+c_{21}) = (c_{11}+c_{22}) \cap ({}_{\max}\{c_{11}, c_{22}\} \le {}_{\max}\{c_{12}, c_{21}\})$ then c_{12}, c_{21} 셀 삭제
  end
  $|c'_{ij}|=0$인 i행이나 j열이 존재하는 c_{ij}에 대해 해당 j열이나 i행의 c'_{ij} 모두 삭제, 이 과정은 조건이 충족되는 한 반복 수행된다.

Step 2. Swap Optimization
  if $|c'_{ij}|=0$ then 알고리즘 종료
  else if $\exists, |c'_{ij}| \ge 2$ then c'_{ij}가 i행에 존재하는 $_{\max}c_{ij}$를 출발지로 하여 $_{\max}c_{ij}$의 j열로 되돌아오는 $\Sigma c_{ij}^0 > \Sigma c_{ij}^+$의 사이클이 형성되도록 $k-opt$ 수행
  endif

Step 2에서의 $k-opt$는 $k=2,3$으로 k개의 셀을 상호 교환 (swap)하는 방식을 의미한다. 제안된 알고리즘은 $n \times n$ 행렬에서 행과 열이 교차하지 않는 n개를 선택하는 방식으로 수행 복잡도는 $O(n^2)$이다.

A_1 할당문제에 대해 제안된 알고리즘은 그림 4와 같이 수행된다. (a)에서, 각 작업 (행) $J_i, i=1,2,\cdots,m$에서의 최소 비용 $_{\min}c_{i\cdot(1)}$을 선택하면 $c_{11}, c_{22}, c_{31}, c_{44}$로 결정된다. 따라서 M_1은 2개 작업으로 과다 선택되었으며, M_3는 선택되지 않아 과소 선택되어 M_1의 J_1, J_3중 어느 하나가 최종적으로 M_3로 이동되어야 한다. 이 는 $(3-1) > (5-4)$로 보다 최소 비용 증가분인 $(J_3, M_1) \rightarrow (J_2, M_2)$로 이동되고, c_{22}, c_{32}중에서 $(10-7) < (11-5)$로 인해 $(J_2, M_2) \rightarrow (J_2, M_3)$로 이동되었다.

다음으로, (b)에서 Phase II의 Step 1 행렬축소를 수행한 결과 교환 최적화 대상인 $c'_{ij} \notin M$이 존재하지 않아 Step 2의 $k-opt$ 교환 최적화를 수행하지 않고 알고리즘이 종료되었다.



(a) One-sided assignment

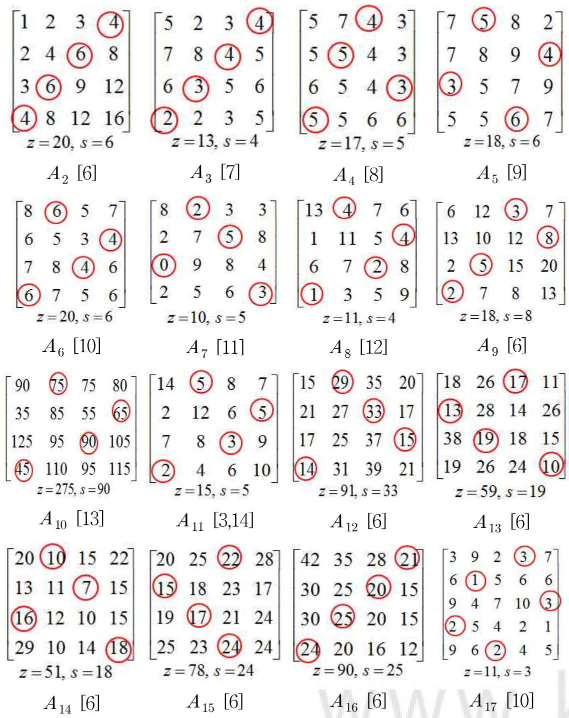
(b) Swap-optimization

Fig. 4. OOAS Algorithm for A_1 Problem

헝가리안 알고리즘은 행과 열의 ${}_{\min}c_i, {}_{\min}c_j$인 최소치 선택과 차이 계산, 선을 긋고, $c_{ij} > 0$에 대해 ${}_{\min}c_{ij}$에 대한 차이 계산과 행과 열의 선이 교차된 c_{ij}에 대한 합 계산 등 복잡한 과정을 수행하며, 마지막으로 행과 열을 중첩되지 않게 1:1로 다시 매칭시켜야 한다. 반면에, 제안된 OOAS 알고리즘은 단순히 행의 단측에 대해서만 최소치를 선택하는 단측-최적화를 수행하고, 과다 선택된 열에서 과소 선택된 열로의 이동을 시킨다. 다음으로, 비용을 감소시킬 수 있는 경우가 발생하는지 검증하기 위해 행렬을 축소시키고, 최적 해를 교환 기법으로 결정한다.

IV. Applications and Evaluation

본 장에서는 그림 5의 26개 균형 할당문제[9-23]를 대상으로 OOAS 알고리즘의 적용성을 평가해 본다. A_{27}은 Dantzig[23]가 해를 제시하지 않아 다른 실험 결과를 인용하였다.



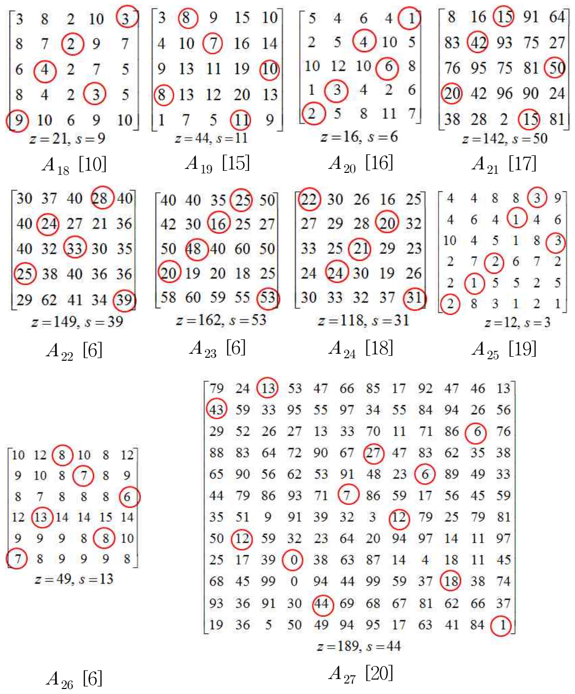
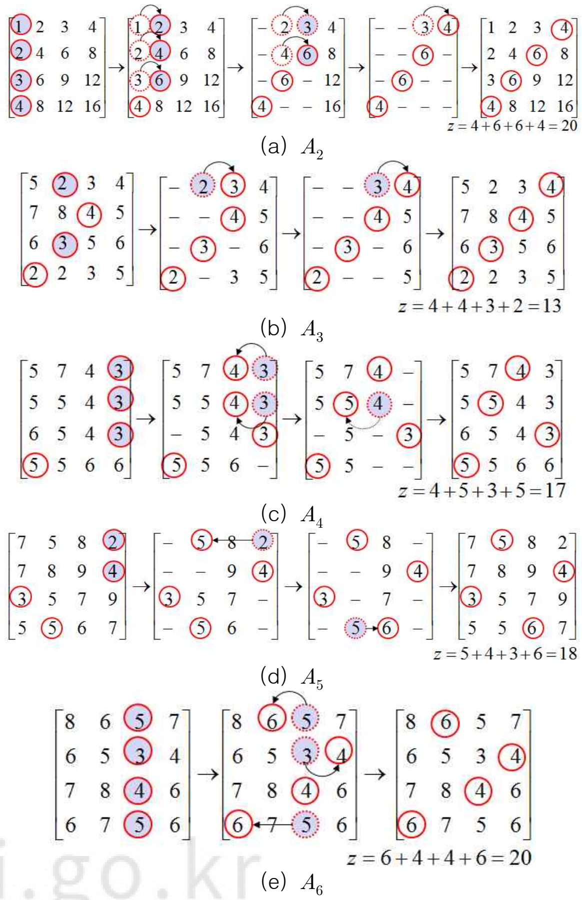
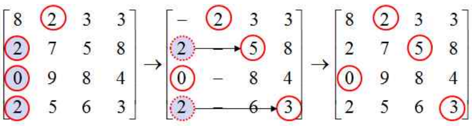


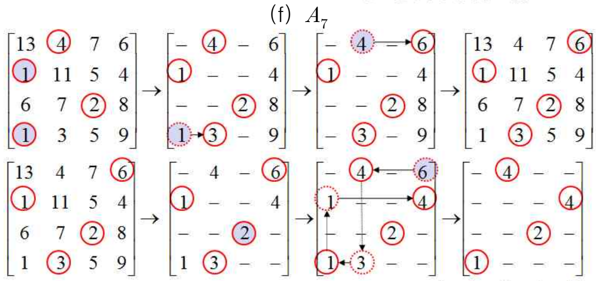
Fig. 5. Experimental Data for Assignment Problems

26개의 균형 할당 문제에 대해, 제안된 OOAS 알고리즘을 적용한 결과는 그림 6에 제시되어 있으며, Phase II의 Step 1 수행 결과 교환 대상이 존재하지 않는 경우는 생략하였다.

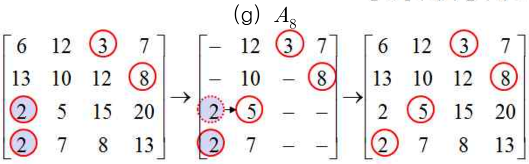




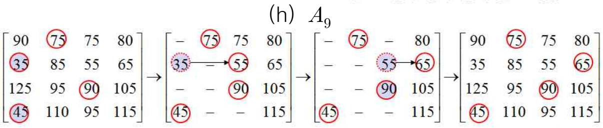
(f) A_7



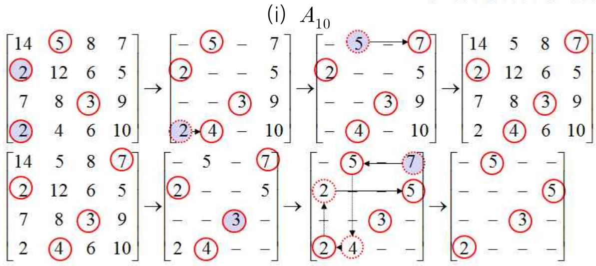
(g) A_8



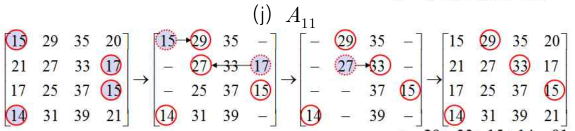
(h) A_9



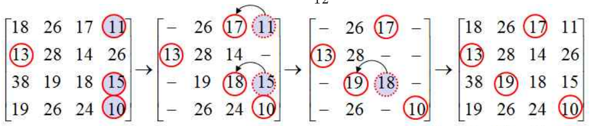
(i) A_{10}



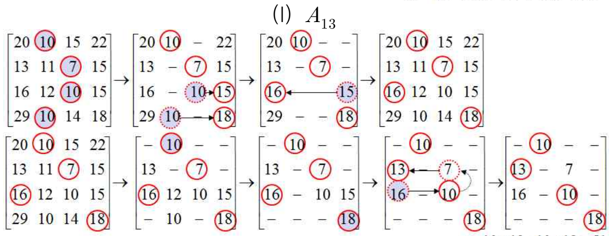
(j) A_{11}



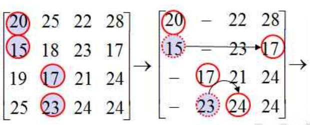
(k) A_{12}



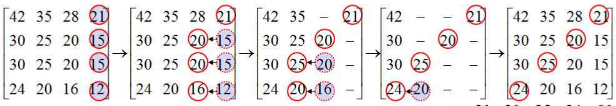
(l) A_{13}



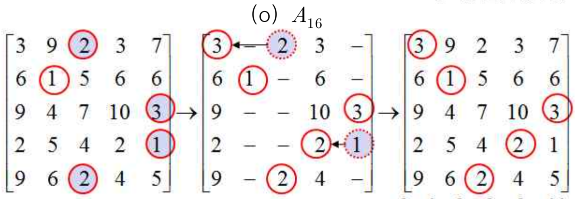
(m) A_{14}



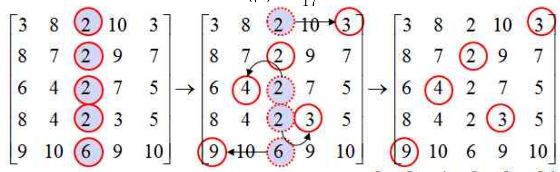
(n) A_{15}



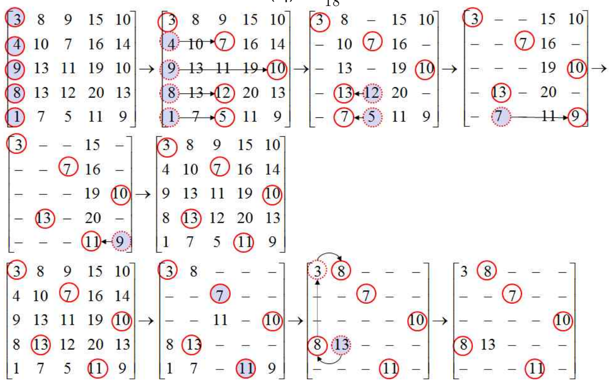
(o) A_{16}



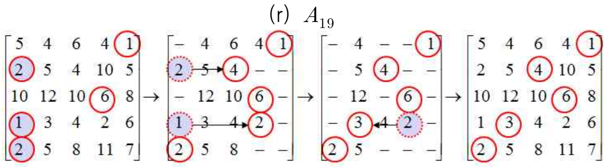
(p) A_{17}



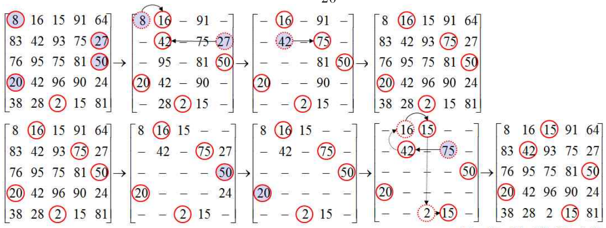
(q) A_{18}



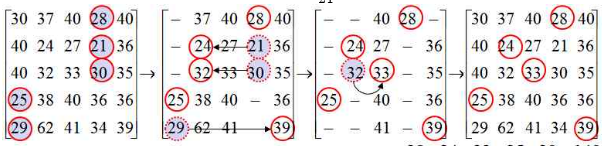
(r) A_{19}



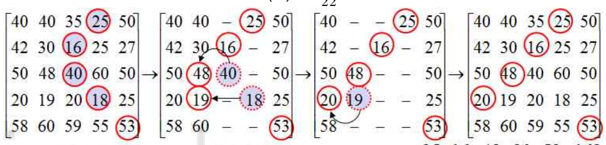
(s) A_{20}



(t) A_{21}



(u) A_{22}



(v) A_{23}

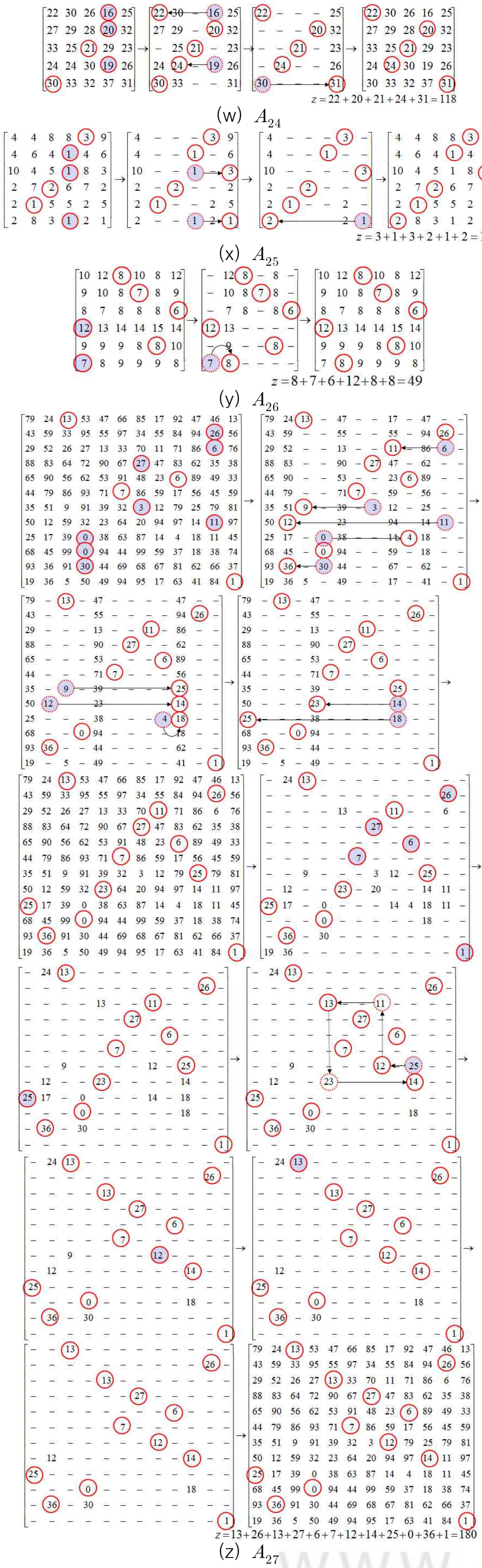


Fig. 6. OOAS Algorithm for Assignment Problem

실험 결과에 대해, 기존에 알려진 최적해와 OOAS의 최적해를 비교한 결과는 표 1에 제시되어 있다. 표에서, OOAS 알고리즘은 $A_8, A_{11}, A_{14}, A_{19}, A_{21}$ 과 A_{27}의 6개 데이터 (22%)에 대해서만 교환 최적화를 수행하였고, 나머지 21개 데이터 (78%)는 단측-최적화 결과로 양측-최적화와 동일한 성능을 얻었음을 알 수 있다. 26개 할당 문제에 대해 제안된 OOAS 알고리즘은 모두 최적 해를 찾았다. 특히, A_{26}은 $_{\min}s$를, A_{27}은 $_{\min}z$와 $_{\min}s$를 구하였다. 여기서 s는 모든 작업이 종료되는 시간으로, 선택된 값들 중 최대값으로 결정된다. 즉, n개의 작업이 n개의 기계에 동시에 투입된다면 s 시간 이후에 제품으로 생산되기 때문에 이를 순환시간이라 하며, 생산관리 분야에서는 제품의 생산성을 결정하는 요인으로 매우 중요하게 고려하고 있다.

Table 1. Compare of algorithm performance

문제	기존에 알려진 해		OOAS		
	z	s	$k-opt$	z	s
A_1	21	10		21	10
A_2	20	6		20	6
A_3	13	4		13	4
A_4	17	5		17	5
A_5	18	6		18	6
A_6	20	6		20	6
A_7	10	5		10	5
A_8	11	4	**3**	11	4
A_9	18	8		18	8
A_{10}	275	90		275	90
A_{11}	15	5	**3**	15	5
A_{12}	91	33		91	33
A_{13}	59	19		59	19
A_{14}	51	18	**2**	51	18
A_{15}	78	24		78	24
A_{16}	90	25		90	25
A_{17}	11	3		11	3
A_{18}	21	9		21	9
A_{19}	44	11	**2**	44	11
A_{20}	16	6		16	6
A_{21}	142	50	**3**	142	50
A_{22}	149	39		149	39
A_{23}	162	53		162	53
A_{24}	118	31		118	31
A_{25}	12	3		12	3
A_{26}	49	**13**		49	**12**
A_{27}	**189**	**44**	**3**	**180**	**36**

제안된 OOAS 알고리즘은 각 행에서의 최소치를 선택하는 단측-최적화를 수행하고, 교환 대상이 존재하는 경우 교환 최적화를 수행하여 $O(n^2)$의 수행 복잡도로 할당문제의 양측-최적화의 목표를 달성할 수 있었다.

결국, OOAS 알고리즘은 헝가리안 알고리즘의 수행 복잡도 $O(n^4)$을 $O(n^2)$으로 향상시켰으며, A_{26} 데이터에 대해서는 순환시간을, A_{27}에 대해서는 순환시간과 작업종료시간의 알려진 최적 해를 개선하는 결과도 얻었다.

V. Conclusions and Future Researches

본 논문에서는 할당 문제의 최적 해를 헝가리안 알고리즘의 $O(n^4)$ 을 $O(n^2)$ 으로 단축시킬 수 있는 알고리즘을 제안하였다. 제안된 알고리즘은 각 행에서 최소 비용을 선택하고, 배정량을 조정하는 단측-최적화를 수행하였다. 다음으로, 교환 대상이 존재하는지 검증하고, 만약, 교환 대상이 존재하면 $k-opt(k=2,3)$ 의 교환 최적화를 수행하여 양측-최적화를 시켜야 하는 할당문제의 최적 해를 구하였다.

제안된 OOAS 알고리즘을 균형 할당 27개 문제에 적용한 결과 모든 문제에 대해 $_{\min}z$와 $_{\min}s$의 최적 해를 찾는데 성공하였다. 또한, 일부 데이터에 대해서는 기존의 알고리즘이 찾은 해를 개선할 수 있었다.

본 논문은 선형 (1차, linear) 할당 문제에 가장 적합한 알고리즘을 제안하였다. 추후 NP-난제로 알려진 2차원 할당문제 (quadratic assignment problem, QAP)에도 본 알고리즘이 적합한지 여부를 검증할 예정이다.

References

[1] E. Budish and E. Cantillon, "The Multi-unit Assignment Problem: Theory and Evidence from Course Allocation at Harvard," American Economic Review, Vol. 102, No. 5, pp. 2237-2271, Aug. 2012.

[2] H. W. Kuhn, "50 Years of Integer Programming 1958-2008, Chapter 2. The Hungarian Method for the Assignment Problem," Springer-Verlag Berlin Heidelberg, pp. 29-47, 2010.

[3] L. Ntaimo, "Introduction to Mathematical Programming: Operations Research: Transportation and Assignment Problems," 4th ed., by W. L. Winston and M. Venkataramanan, 2005.

[4] K. Kinahan and J. Pryor, "Algorithm Animations for Practical Optimization: A Gentle Introduction," Systems and Computer Engineering, Carleton University, Ottawa, Ontario, Canada, 2007.

[5] D. N. Kumar, "Optimization Methods," http://www.nptel.iitm.ac.in/ourses/Webcourse-contents/IISc-BANG/OPTIMIZATION METHODS/pdf/Optimization_Methods_Syllabus.pdf, IISc, Bangalore, 2008.

[6] x-ray-TopCoder "Assignment Problem and Hungarian Algorithm," https://www.topcoder.com/community/data-science/data-science-tutorials/assignment-problem-and-hungarian-algorithm/, Topcoder, 2014.

[7] S. Singh, "Note on Assignment Algorithm with Easy Method of Drawing Lines to Cover All Zeros," International Journal of Operations Research and Information Systems, Vol. 3, No. 3, pp. 87-97, Jul. 2012.

[8] S. U. Lee, "An Assignment Problem Algorithm Using Minimum Cost Moving Method," Journal of KSCI, Vol. 20, No. 8, pp. 105-112, Aug. 2015.

[9] Rai Foundation Colleges, "Information Research," Bachelor of Business Administration, Business Administration, 2008.

[10] S. Noble, "Lectures 15: The Assignment Problem," Department of Mathematical Sciences, Brunel University, 2000.

[11] A. Dimitrios, P. Konstantinos, S. Nikolaos, and S. Angelo, "Applications of a New Network-enabled Solver for the Assignment Problem in Computer-aided Education," Journal of Computer Science, Vol. 1, No. 1, pp. 19-23, Jan. 2005.

[12] R. M. Berka, "A Tutorial on Network Optimization," http://www.berkovi.cz/milan/berka/o/English/networks/networks.html, 1997.

[13] M. S. Radhakrishnan, "AAOC C222: Optimization," Birla Institute of Technology & Science, 2006.

[14] R. Burkard, M. D. Amico, and S. Martello, "Assignment Problems, SIAM Monographs on Discrete Mathematics and Applications, 2006.

[15] S. C. Niu, "Introduction to Operations Research," School of Management, The University of Texas at Dallas, 2004.

[16] W. Snyder, "The Linear Assignment Problem," Department of Electrical and Computer Engineering, North Carolina State University, 2005.

[17] M. E. Salassi, "AGEC 7123: Operations Research Methods in Agricultural Economics: Standard LP Form of the Generalized Assignment Problem," Department of Agricultural Economics and Agribusiness, Louisiana State University, 2005.

[18] K. Wayne, "Algorithm Design," http://www.cs.princeton.edu/~wayne/kleinberg-tardos/07assignment.pdf, 2005.

[19] J. Havlicek, "Introduction to Management Science and Operation Research," http://orms.czu.cz/text/transproblem. html, 2007.

[20] R. Sedgewick and K. Wayne, "Computer Science 226: Data Structures and Algorithms, Princeton University, 2002.

[21] J. E. Beasley, "Operations Research and Management

Science: OR-Notes," Department of Mathematical Sciences, Brunel University, West London, 2004.
[22] D. Doty, "Munkres' Assignment Algorithm: Modified for Rectangular Matrices," KCVU, Murray State University, Dept. of Computer Science and Information Systems, 2008.
[23] G. B. Dantzig, "Linear Programming and Extensions," USAF Project RAND, R-366-PR, The RAND Corporation, Santa Monica, California, U.S., 1963.

Authors

Sang Un Lee received the B. Sc. degree in avionics from the Korea Aerospace University in 1997. He received the M. Sc. and Ph. D. degrees in Computer Science from Gyeongsang National University, Korea, in 1997 and 2001, respectively. He is currently Professor with the Department of Multimedia Science, Gangneung-Wonju National University, Korea. He is interested in software quality assurance and reliability modeling, software engineering, software project management, neural networks, and algorithm.