Journal of Engineering, Project, and Production Management
2022, 12(3), 209-216

Approach for Solving Project Assignment with Bicriterion

Ding-Tsair Chang[1], Hsien-Hong Lin[2], Su-Hui Chen[3], and Chiu-Chi Wei[4]
[1]Associate Professor, Department of Industrial Management, Chung Hua University, No. 707, Sec. 2, Wufu Rd., HsinChu, Taiwan, E-mail: apma.service@gmail.com
[2]Associate Professor, Department of Logistics Management, Huaiyin Institute of Technology, No.1, Mei Chen E. Rd., Huaiyin, Chian Su, China, E-mail: joelin2005usa@yahoo.com.tw
[3]Associate Professor, Department of Human Resources Management, Huaiyin Normal University, No. 111, Chang Chian W. Rd., Huaiyin Dist. Huaian, Chian Su, China, E-mail: 8201901026@hytc.edu.cn
[4]Professor, Department of Industrial Management, Chung Hua University, No. 707, Sec. 2, Wufu Rd., HsinChu, Taiwan
E-mail: a0824809@gmail.com (corresponding author).

Project Management
Received August 11, 2021; revised February 5, 2022; February 14, 2022; accepted February 20, 2022
Available online May 6, 2022

Abstract: Assigning the right member to the most suitable position is key to the success of a project, and this task has been commonly executed by the project manager based on personal judgment in practice. This paper proposes a matching approach coupled with a revised Hungarian algorithm for optimizing the cost-time project assignment problem. The approach iteratively searches for the augmenting path concerning the current matching rather than solving the entire problem repeatedly. This unique feature greatly reduces the computation efforts. Problems of different sizes and sample ranges are simulated using the proposed technique and G&N's method. Results show that the presented algorithm excels the previous approach in not only producing a lower bound for the project time but also in reaching the optimal solution using much less computing time.

Keywords: project assignment, cost-minimizing, time-minimizing, matching.

DOI 10.32738/JEPPM-2022-0019

1. Introduction

Deploying N resources to N tasks with various consumption rates is a common task in project management and is widely known as the assignment problem. Since budgeted resources are frequently limited in nature, it is necessary to allocate scarce resources efficiently. Therefore, the problem is traditionally formulated as assigning the optimal available resources to the most needed tasks so as to minimize and maximize the project cost and benefit, respectively. Problems of this type can be the deployment of N workers to N jobs, N machines to N operations, or N routes to N city buses (Shopov and Markova, 2021; Gabrovšek et al., 2020). Conventional treatment of the problem falls into one of the following categories: (1) cost-minimizing assignment problems (Barr et al. 1977; Hung and Rom, 1980), and (2) time minimizing assignment problems (Garfinkel, 1971). Cost-oriented problems aim at finding the optimal assignment that minimizes the total project cost, whereas time-oriented problems focus on searching for the shortest duration when the total project time is of vital concern. Generalizations of the cost minimization and bottleneck assignment problems have been discussed in the literature (Fisher et al., 1986; Mazzola and Neebe, 1993; Mazzola, 1989; Mazzola and Neebe, 1988; Ross and Soland, 1977; Degroote et al., 2018; Karsu and Azizoğlu, 2012; Moussavi et al., 2018; Posta et al. 2012; Wu et al. 2018).

The major characteristic of the project assignment problem is that the number of jobs and workers are identical, while it can be viewed as a special type of transportation problem in which the number of demands and suppliers are usually different. Therefore, the assignment problem is alternatively called the balanced transportation problem. Previous work used the transportation method, such as least cot, north-west corner, and Vogel's approximation (Das et al., 2014; Palaniyappa, 2016; Prasad and Singh, 2020) and the simplex method (Arsham and Kahn, 1989; Bulut, 2016) to solve the assignment problems, however, the high degree of degeneracy leads to the methods inefficient. The Hungarian method is another more efficient and widely applied technique to optimize the assignment (Munapo, 2020; Khan et al., 2020; Gabrovšek et al., 2020). The above methods assume that the optimal assignment can be reached if the single objective of either total project cost or overall project time is minimized. Unfortunately, cost and time are frequently correlated. Thus, separating cost-minimizing assignments from the time minimizing assignments does not correctly reflect the real situations. It is easily realized that the amount of cost needed and time

required to accomplish a task are highly interrelated with the difficulty of the job and the competency of the worker. It is actually a trade-off between time and cost.

In 1993, Geetha and Nair (G&N's) extended the assignment to a multi-objective problem associated with time and cost and proposed an algorithm to solve it. Due to the fact that the longer the project, the more effort the management must devote. Therefore, the supervisory cost per unit of time with respect to the project duration was introduced. The decision to be made is to minimize the total project cost plus the cost of supervising the most time-consuming job. This study adopts G&N's definition and proposes a more efficient approach for optimizing the project cost-time assignment problem.

2. Problem Formulation

Two major assumptions related to the project assignment problems are (1) each worker i is assigned exactly to one job j and each job j is assigned exactly to one worker i, and (2) all jobs are undertaken simultaneously. Thus the cost-minimizing assignment can be stated as Eq. (1):

$$MinC = \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij} \quad (1)$$

subject to

$$\sum_{j=1}^{n} X_{ij} = 1 \quad \text{for i=1, 2, ..., n}$$

$$\sum_{i=1}^{n} X_{ij} = 1 \quad \text{for j=1, 2, ..., n}$$

$X_{ij} = 1$ if job j is assigned to worker i;

$= 0$ otherwise.

where C_{ij} is the cost of assigning job j to worker i.

Based on the assumptions mentioned above, the total project duration is defined as the minimum completion time required to complete all the jobs. Then the time minimizing assignment can be denoted as Eq. (2) (Geetha and Nair, 1993):

$$MinT = Max\{t_{ij} | x_{ij} = 1\} \quad (2)$$

where t_{ij} is the time needed for the worker i to complete job j.

A longer project time mostly implies a higher project cost due to extra costs incurred in association with various activities such as the supervisory efforts from the management and support from the indirect employee. It is, therefore, possible to convert the project duration into project cost by multiplying the suitable unit cost. Hence, the completion time of the tasks in this study can be added to the cost (Geetha and Nair, 1993). The problem then becomes finding the minimum value of the following problem shown as Eq. (3).

$$MinZ = \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij} + Max\{t_{ij} | x_{ij} = 1\} \quad (3)$$

subject to

$$\sum_{j=1}^{n} X_{ij} = 1 \quad \text{for i = 1, 2, ..., n}$$

$$\sum_{i=1}^{n} X_{ij} = 1 \quad \text{for j = 1, 2, ..., n}$$

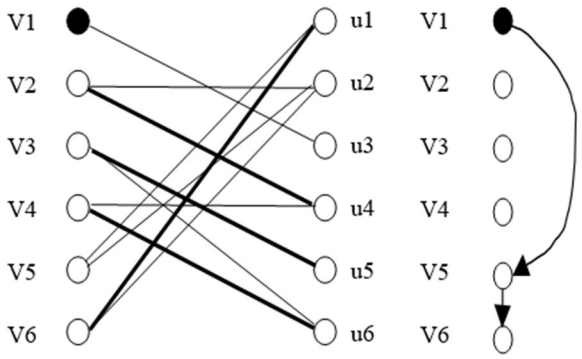
$X_{ij} = 1$ if job j is assigned to worker i;

$= 1$ otherwise.

3. The Proposed Algorithm

The project assignment problem can be represented by a weighted bipartite matching graph G=(W, E) where W is a finite set of nodes or vertices and can be partitioned into two sets, V and U, and E has elements subsets of W of cardinality two called edges. A matching M of a graph G is a subset of the edges with the property that no two edges of M share the same node. Given a graph G=(V, E) the matching problem is to find a maximum matching M of G. Edges in M are called matched; the other edges are free. If $[u, v]$ is a matched edge, then u is the mate of v; nodes that are not incident upon any matched edge are called exposed; the remaining nodes are matched. A path $p=[u_1, u_2, ..., u_k]$ is called alternating if the edges $[u_1, u_2], [u_3, u_4], ..., [u_{2j-1}, u_{2j}], ...$ are free, whereas $[u_2, u_3], [u_4, u_5], ..., [u_{2j}, u_{2j+1}], ...$ are matched. An alternating path $p = [u_1, u_2, ..., u_k]$ is called augmenting path if both u_1 and u_k are exposed vertices. Vertices that lie on an alternating path starting with an exposed vertex and have an odd rank on this path are called outer, the other vertices with an even rank are called inner.

In Fig. 1(a), $p_2 = [v_1, u_3, v_5, u_2]$ is an augmenting path, where v_1 and u_2 are exposed vertices, $[u_3, v_5]$ is a matched edge, v_5 is a mate of u_3, u_3 is an inner vertex and v_5 is an outer vertex, and $[v_2, u_2]$ is a free edge. The significance of augmenting paths for the matching problem is due to the following facts (Papadimitriou and Steiglitz, 1982):



(a) a bipartite graph (b) an auxiliary graph

Figure 1. An illustrative bipartite matching graph

Lemma 1: Let P be the set of edges on an augmenting path $p=[u_1, u_2, \ldots, u_{2k}]$ in a graph G with respect to the matching M, then M'=(M-P) ∪ (P-M) is a matching of cardinality $| M | + 1$.

Lemma 2: A matching M in a graph G is maximum if and only if there is no augmenting path in G with respect to M.

The breadth-first search can be applied to search augmenting paths in a graph, however, due to its special structure, we can simplify this searching technique by ignoring the odd-numbered levels and going directly from the outer vertices to new outer vertices. An auxiliary digraph, which contains just vertices of V can be constructed because only vertices of V can become outer vertices in alternating paths starting from V. The auxiliary graph corresponding to Fig. 1(a) is shown in Fig. 1(b), it is noted that breadth-first search of the auxiliary graph starting from the exposed node v_1 obtains an augmenting path $[v_1, u_3, v_5, u_2]$. Furthermore, $[v_1, u_3, v_5, u_1, v_6, u_2]$ is another augmenting path.

For the assignment problem (completed weighted bipartite matching problem), we can formulate the primal problem (*P*) as Eq. 4:

$$(P) \qquad Min \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij} \quad (4)$$

subject to

$$\sum_{j=1}^{n} X_{ij} \qquad \forall i$$

$$\sum_{i=1}^{n} X_{ij} \qquad \forall j$$

$$X_{ij} \geq 0 \qquad \forall ij$$

The corresponding dual problem (D) can be described as Eq. 5:

$$(D) \qquad Max \sum_{i=1}^{n} \alpha_i + \sum_{j=1}^{n} \beta_i \qquad (5)$$

subject to

$$\alpha_i + \beta_j \geq C_{ij}$$

where α_i and β_j are unrestricted, and X_{ij}=1 means that the edge [v_i, u_j] is included in the matching, and x_{ij}=0 otherwise. An initial feasible solution to the dual problem can be obtained as:

$$\alpha_i = 0$$

$$\beta_j = \min\{C_{ij}\}$$

In order to apply the primal-dual algorithm to the assignment problem, the corresponding restricted primal problem (*RP*) can be written as Eq. 6:

$$Min \sum_{i=1}^{n} x_i^a + \sum_{j=1}^{n} y_j^a \qquad (6)$$

subject to:

$$x_{ij} \geq 0 \qquad \text{if } (i,j) \in IJ$$

$$\sum_{j=1}^{n} x_{ij} + x_i^a = 1 \quad \forall i$$

$$x_{ij} = 0 \qquad \text{if } (i,j) \notin IJ$$

$$\sum_{i=1}^{n} x_{ij} + y_j^a = 1 \quad \forall j$$

$$x_i^a \geq 0, y_j^a \geq 0$$

where *IJ* is the admissible set defined as

$$IJ = \{(i, j) \mid \alpha_I + \beta_j = C_{ij}\}.$$

Furthermore, replacing x_i^a by $1 - \sum_{j=1}^{n} X_{ij}$ and y_j^a by $1 - \sum_{j=1}^{n} X_{ij}$, then *RP* is equivalent to *RP'* shown as Eq. 7:

$$(RP') \qquad Max \sum_{(i,j)\in IJ} x_{ij} \qquad (7)$$

subject to

$$\sum_{j=1}^{n} x_{ij} \leq 1 \qquad \forall i$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad \forall j$$

$$x_{ij} \geq 0 \text{ if } (i,j) \in IJ$$

$$x_{ij} = 0 \text{ if } (i,j) \notin IJ$$

The dual problem of *RP'* (*DRP'*) is formulated as Eq. 8:

$$(DRP') \quad Min \sum_{i=1}^{n} \alpha_i + \sum_{j=1}^{n} \beta_j \qquad (8)$$

subject to

$$\alpha_i + \beta_j \geq 1 \qquad \text{if } (i,j) \in IJ$$

$$\beta_j \geq 0$$

$$\alpha_i + \beta_j \geq 0 \qquad \text{if } (i,j) \notin IJ$$

$$\alpha_i \geq 0$$

It is easy to see that RP is an unweighted bipartite matching problem for the bipartite graph consisting of currently admissible matching edges. While searching for an augmenting path, it may be possible that no augmenting path is in the present set of admissible edges. Then, we shall change the dual variables α_i and β_j to make new edges admissible and resume the search for an augmenting path. To modify the dual variables, we need to calculate the following:

$$\theta_1 = \frac{1}{2} \min_{ij}\{c_{ij} - \alpha_i - \beta_j\}$$

The minimum is taken over all labeled nodes $v_i \in V$ and unlabeled nodes $u_j \in U$, and the optimal solution is achieved when the bipartite graph is completely matched. Combining the matching technique with the Hungarian method (Papadimitriou and Steiglitz, 1982), the following algorithm can be derived to solve the bicriterion assignment problem.

Algorithm:

Step 0: Set iteration number k = 1, number of matches m = 0, n = $|V|$, $Q^* = \infty$, find the lower bound b for Z_2=max$\{t_{ij} | X_{ij} = 1\}$, and construct the initial graph for the assignment.

Step 1: construct an auxiliary graph for the bipartite graph of the assignment problem using the Hungarian method.

Step 2: find an augmenting path then go to step 4, otherwise go to Step 3.

Step 3: modify the auxiliary graph, go to step 2.

Step 4: a new match is found, set m = m + 1. If m = n then go to step 5, otherwise go to step 1.

Step 5: a feasible solution is found, let $Z_1^k = \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij} X_{ij}$, $Z_2^k = max\{ t_{ij} | X_{ij} = 1\}$, $Q^k = Q_1^k + Z_2^k$. If $Q^k < Q^*$ then set $X^* = X^k, Q^* = Q^k, Z_1^* = Z_1^k, Z_2^* = Z_2^k$, go to step 6. Otherwise, if $Z_1^k \geq Q^* - b$, then stop, in this case, X^* is an optimal solution, and Q^* is the optimal value.

Step 6: Revise C_{ij}s' as follow:

$$C_{ij} = \begin{cases} C_{ij} & if \quad t_{ij} < Z_2^k \\ \infty & if \quad t_{ij} \geq Z_2^k \end{cases}$$

Step 7: Remove all arcs (i, j) of the bipartite graph for which $C_{ij} = \infty$. Set m = m - matches removed; set k = k + 1 and go to Step 1.

Note that the lower bound b of time is obtained using the procedures listed below. Let $r_i = \min_i\{t_{ij}\}$, $c_j = \min_j\{t_{ij}\}$, $r = \max_i\{r_i\}$, $c = \max_j\{c_j\}$, then b = max {r, c}.

It can be seen that, compared to previous work by G&N, a better value of lower bound can be generated, which results in the reduction of the feasible set and the searching time. Step 7 in the proposed algorithm will be performed at most n^2 times and the complexity of the Hungarian Method utilized in the algorithm is $O(n^3)$. Hence the complexity of the proposed algorithm will be $O(n^5)$ for the worst case. However, in the general case, the performance of the algorithm will be better than $O(n^5)$.

4. Illustrative Example

The algorithm proposed is applied to find the optimal cost-time assignment problem given in Table 1 (Mazzola and Neebe, 1993). Apparently, six jobs are to be assigned to 6 workers with different costs and time. To clearly demonstrate the solution processes of the algorithm, the problem is solved step by step with the matching graph depicted (see Table 2).

Table 1. The cost-time assignment problem

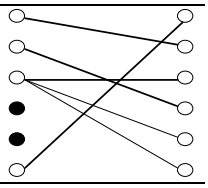
			Job j					
			1	2	3	4	5	6
Worker i	1	C_{ij}	6	3	5	8	10	6
		t_{ij}	4	20	9	3	8	9
	2	C_{ij}	6	4	6	5	9	8
		t_{ij}	6	18	8	7	17	8
	3	C_{ij}	11	7	4	8	3	2
		t_{ij}	2	8	20	7	15	7
	4	C_{ij}	9	10	8	6	10	4
		t_{ij}	12	13	14	6	9	10
	5	C_{ij}	4	6	7	9	8	7
		t_{ij}	9	8	7	14	5	9
	6	C_{ij}	3	5	11	10	12	8
		t_{ij}	17	13	3	4	13	7

In table 2, * sign indicates a labeled but unmatched node, $\tilde{\mathbf{3}}$ is a possible matching edge (admissible edge); $\boxed{3}$ is a matched edge. Moreover, in the column of the label, -1 means a starting node, and 0 means unlabeled. In the column of exposed, 0 implies no exposed node connected. Slack is computed as slack[u_j]=min{C_{ij}-α_i-β_j}, nhbor[u_j] is the particular labeled vertex v_i with which slack[u_j] is achieved, and match[u_j] is the particular vertex v_i matches with u_j. In the bipartite graph below the tables, ● is an unmatched node, and ○ is a matched node; a bold line represents a matched edge, and a plain line is an admissible edge.

Table 2. Solution processes

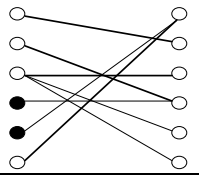
(a) stage 1

	1	2	3	4	5	6	α	exposed	label
1	6	**3**	5	8	10	6	0	0	0
2	6	4	6	**5**	9	8	0	0	0
3	11	7	**4**	8	$\tilde{3}$	$\tilde{2}$	0	5	0
* 4	9	10	8	6	10	4	0	0	-1
* 5	4	6	7	9	8	7	0	0	-1
6	**3**	5	11	10	12	8	0	0	0
β	3.0	3.0	4.0	5.0	3.0	2.0			
slack	1.0	3.0	3.0	1.0	5.0	2.0			
nhbor	5	5	5	4	5	4			
match	2	4	3	0	0	1			



(b) stage 2

	1	2	3	4	5	6	α	exposed	label
1	6	$\boxed{3}$	5	8	10	6	-0.5	0	0
v 2	6	4	6	$\boxed{5}$	9	8	-0.5	0	4
3	11	7	$\boxed{4}$	8	$\tilde{3}$	$\tilde{2}$	-0.5	5	0
* 4	9	10	8	$\tilde{6}$	10	4	-0.5	0	-1
* 5	$\tilde{4}$	6	7	9	8	7	-0.5	0	-1
v 6	**3**	5	11	10	12	8	-0.5	0	5
β	3.5	3.5	4.5	5.5	3.5	2.5			
slack	0.0	1.0	2.0	0.0	4.0	1.0			
nhbor	5	2	2	2	5	4			
match	2	4	3	0	0	1			



v: labeled node and matched

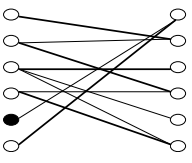
(c) stage 3

	1	2	3	4	5	6	α	exposed	label
v 1	6	$\boxed{3}$	5	8	10	6	-1.0	0	2
v 2	6	$\tilde{4}$	6	$\boxed{5}$	9	8	0.0	0	4
3	11	7	$\boxed{4}$	8	$\tilde{3}$	$\tilde{2}$	-1.0	5	0
* 4	9	10	8	$\tilde{6}$	10	$\tilde{4}$	1.0	0	-1
* 5	$\tilde{4}$	6	7	9	8	7	1.0	0	-1
v 6	**3**	5	11	10	12	8	-0.5	0	5
β	3.0	4.0	5.0	5.0	4.0	3.0			
slack									
nhbor									
match	2	4	3	0	0	1			

An augmenting path [v_4, u_6] is found.

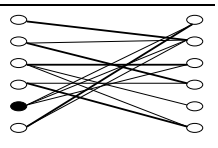
(d) stage 4

	1	2	3	4	5	6	α	exposed	label
1	6	$\boxed{3}$	5	8	10	6	-1.0	0	2
2	6	$\tilde{4}$	6	$\boxed{5}$	9	8	0.0	0	4
3	11	7	$\boxed{4}$	8	$\tilde{3}$	$\tilde{2}$	-1.0	5	0
4	9	10	8	$\tilde{6}$	10	$\boxed{4}$	1.0	0	0
*5	$\tilde{4}$	6	7	9	8	7	1.0	0	-1
v 6	**3**	5	11	10	12	8	0.0	0	5
β	3.0	4.0	5.0	5.0	4.0	3.0			
slack	0.0	1.0	1.0	3.0	3.0	3.0			
nhbor	5	5	5	5	5	5			
match	2	4	3	6	0	1			



(e) stage 5

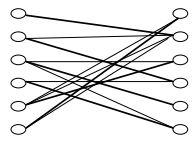
	1	2	3	4	5	6	α	exposed	label
v 1	6	**3**	5	8	10	6	-1.0	0	2
v 2	6	$\underline{\tilde{4}}$	6	**5**	9	8	0.0	0	4
v 3	11	7	**4**	8	$\tilde{3}$	$\underline{\tilde{2}}$	-1.0	5	0
v 4	9	10	8	$\underline{\tilde{6}}$	10	**4**	1.0	0	0
*5	$\underline{\tilde{4}}$	$\underline{\tilde{6}}$	$\tilde{7}$	9	8	7	1.0	0	-1
v 6	**3**	$\underline{\tilde{5}}$	11	10	12	8	0.0	0	5
β	2.5	4.5	5.5	5.5	4.5	3.5			
slack									
nhbor									
match	2	4	3	6	0	1			



An augmenting path $[v_5, u_3, v_3, u_5]$ is found

(f) stage 6

	1	2	3	4	5	6	α	exposed	label
1	6	**3**	5	8	10	6	-1.5		
2	6	$\underline{\tilde{4}}$	6	**5**	9	8	-0.5		
3	11	7	**4**	8	3	$\underline{\tilde{2}}$	-1.5		
4	9	10	8	$\underline{\tilde{6}}$	10	**4**	0.5		
5	$\underline{\tilde{4}}$	$\underline{\tilde{6}}$	7	9	8	7	1.5		
6	**3**	$\underline{\tilde{5}}$	11	10	12	8	0.5		
β	2.5	4.5	5.5	5.5	4.5	3.5			
slack									
nhbor									
match	2	4	3	6	3	1			



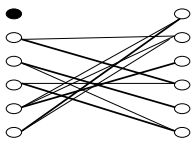
$X^1 = \{x_{12}, x_{24}, x_{35}, x_{46}, x_{53}, \boxed{x_{61}}\}$,

$Z_1^1 = 25,\ Z_2^1 = 20,$

$Z_1^1 + Z_2^1 = 45,\ Q^* = 45$

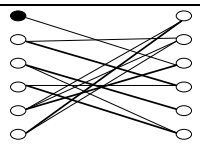
(g) stage 7

	1	2	3	4	5	6	α	exposed	label
* 1	6	∞	5	8	10	6	-1.5	0	-1
2	6	$\underline{\tilde{4}}$	6	**5**	9	8	-0.5	2	0
3	11	7	∞	8	3	$\underline{\tilde{2}}$	-1.5	0	0
4	9	10	8	$\underline{\tilde{6}}$	10	**4**	0.5	0	0
5	$\underline{\tilde{4}}$	$\underline{\tilde{6}}$	7	9	8	7	1.5	2	0
6	**3**	$\underline{\tilde{5}}$	11	10	12	8	0.5	2	0
β	2.5	4.5	5.5	5.5	4.5	3.5			
slack	5.0	∞	1.0	4.0	7.0	4.0			
nhbor									
match	1	1	1	1	1	1			



(h) stage 8

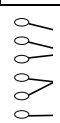
	1	2	3	4	5	6	α	exposed	label
* 1	6	3	$\underline{\tilde{5}}$	8	10	6	-1.0	0	-1
2	6	$\underline{\tilde{4}}$	6	**5**	9	8	-1.0	2	0
3	11	7	4	8	3	$\underline{\tilde{2}}$	-2.0	0	0
4	9	10	8	6	10	4	0,0	0	0
5	$\underline{\tilde{4}}$	$\underline{\tilde{6}}$	7	9	8	7	1,0	2	1
6	**3**	$\underline{\tilde{5}}$	11	10	12	8	0,0	2	5
β	3.0	5.0	6.0	6.0	5.0	4.0			
slack									
nhbor									
match	3	4	5	6	2	1			



An augmenting path $[v_1, u_3, v_5, u_2]$ is found

(i) final stage

	1	2	3	4	5	6	α	exposed	label
1	6	∞	5	8	10	6	-0.5	0	0
2	$\underline{\tilde{6}}$	∞	$\underline{\tilde{6}}$	**5**	∞	8	-0.5	0	4
3	11	7	∞	8	∞	$\underline{\tilde{2}}$	-0.5	5	0
4	9	∞	∞	6	10	4	-0.5	0	-1
5	4	$\underline{\tilde{6}}$	7	∞	$\underline{\tilde{8}}$	7	-0.5	0	-1
6	∞	∞	$\underset{11}{\tilde{}}$	$\underset{10}{\tilde{}}$	∞	8	-0.5	0	5
β	3.5	3.5	4.5	5.5	3.5	2.5			
slack	0.0	1.0	2.0	0.0	4.0	1.0			
nhbor	5	2	2	2	5	4			
match	2	4	3	0	0	1			



$X^5 = \{x_{13}, x_{24}, x_{32}, x_{45}, x_{51}, x_{66}\}$,

$Z_1^5 = 39,\ Z_2^5 = 9,\ Z_1^5 + Z_2^5 = 48,\ Q^* = 41.$

For simplicity, the detailed calculation procedures of the Hungarian method will be omitted. In the first stage shown in Table 2(a), set $\alpha_i = 0$, $\beta_j = \min_i\{C_{ij}\}$, compute the lower bound of time b, and mark the admissible edges for matching, we can obtain four matches, $[v_1, u_2]$, $[v_2, u_4]$, $[v_3, u_3]$ and $[v_6, u_1]$. Unmatched nodes v_4 and v_5 are labeled with -1. Node v_3 can be connected to the exposed nodes u_5 and u_6, u_5 is selected in this case. Because no augmenting path is possible, go to Step 3. The smallest nonzero slack in Table 2(a) is 1.0, therefore, $\theta_1 = 0.5$. The revised Table 2(b) shows that edges $[v_4, u_4]$ and $[v_5, u_1]$ can be added to the graph and no edges are deleted, go to Step 3 with $\theta_1 = 0.5$, the results are shown in Table 2(c). Clearly, edges $[v_2, u_2]$ and $[v_4, u_6]$ can be added to the graph and no edges are deleted. In this stage, an augmenting path $[v_4, u_6]$ is found. Thus graph is modified as Table 2(d). Since no augmenting path is possible, go to Step 3 with $\theta_1 = 0.5$. The new graph in Table 2(e) shows that edges $[v_5, u_2]$, $[v_5, u_3]$, and $[v_6, u_2]$ can be added to the graph and no edges are deleted, an augmenting path $[v_5, u_3, v_3, u_5]$ is found here. The graph is changed to Table 2(f).

A local optimal is obtained with cost $Z_1^1 = 25$, $Z_2^1 = 20$ and $Q^1 = 45$. Therefore, let X* = X1, Q* = Q1 = 45, $Z_1^* = Z_1^1 = 25$, and $Z_2^* = Z_2^1 = 20$. Go to Step 1 and revise C_{ij}s for the graph. Note that the edges [v_1, u_2] (matched edge) and [v_3, u_3] (admissible edge) are removed, the new graph is shown in Table 2 (g). Because no augmenting path is possible, go to step 3 with $\theta_1 = 0.5$. The modified results are shown in table 2(h), and the edge [v_1, u_3] is added. Similar procedures can be carried out until the global solution is obtained. The results of the last stage are shown in Table 2(i), which shows that $Z_1^5 = 39$, $Z_2^5 = 39$ and $Q^5 = 48$. The local optimal found in each iteration is listed in table 3, where b=8, and # indicates global optimal. Note that the global optimal is obtained in iteration 3.

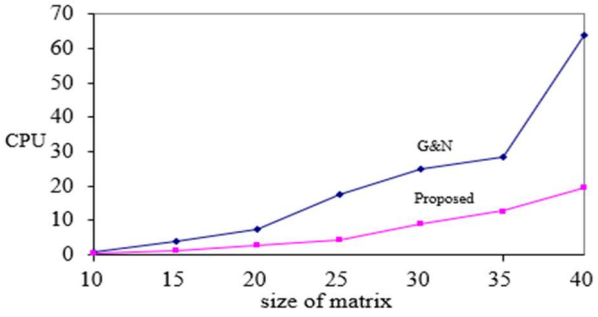
Table 3. The local optimal found in each stage

k	Solution x_{ij}	Z_1^k	Z_2^k	$Z_1^k + Z_2^k$	Q^*	$Z_1^k \geq Q^* - b$
1	12, 24, 35, 46, 53, 61	25	20	45	45	No
2	13, 24, 35, 46, 52, 61	26	17	43	43	No
3	13, 24, 35, 46, 51, 62	26	15	41	#41	No
4	13, 24, 36, 45, 51, 62	31	13	44	41	No
5	13, 24, 32, 45, 51, 66	39	9	48	41	Yes

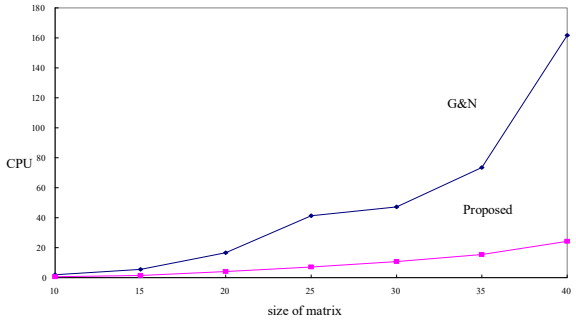
Table 4. Comparison of the results

Range	n	G&N method					Proposed method				
		C	T	Z	b	CPU	C	T	Z	b	CPU
lower	**10**	119	19	138	11	0.99	119	19	138	13	0.55
=10	15	176	20	196	11	3.96	176	20	196	13	1.15
	20	224	20	244	11	7.36	224	20	244	12	2.64
upper	25	277	20	297	11	17.52	277	20	297	12	4.34
=20	30	334	20	354	11	24.93	334	20	354	12	9.07
	35	386	20	406	11	28.51	386	20	406	12	12.69
	40	443	17	460	11	63.77	443	17	460	12	19.55
lower	10	150	37	187	12	1.87	150	37	187	17	0.66
=10	15	211	38	249	12	5.49	211	38	249	19	1.49
	20	248	39	287	12	16.59	248	39	287	15	4.12
upper	25	304	40	344	12	41.30	304	40	344	16	7.09
=40	30	360	40	400	12	47.19	360	40	400	14	10.76
	35	409	40	449	12	73.54	409	40	449	15	15.43
	40	474	37	511	11	161.70	474	37	511	14	24.16
lower	10	212	71	283	15	2.41	212	71	283	27	0.72
=10	15	283	78	361	14	11.26	283	78	361	30	1.81
	20	299	77	376	14	20.92	299	77	376	21	5.06
upper	25	367	69	436	12	28.89	367	69	436	22	7.15
=80	30	417	74	491	12	61.24	417	74	491	18	14.50
	35	449	68	517	11	119.90	449	68	517	16	20.93
	40	509	78	587	11	252.90	509	78	587	16	34.06

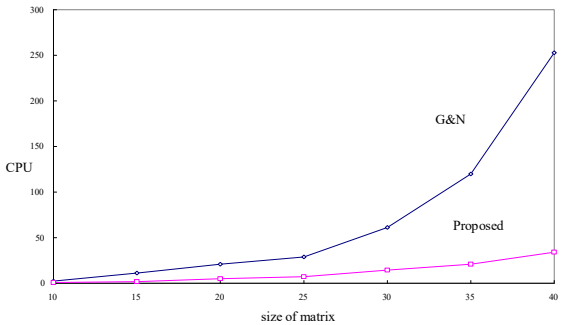
To further verify the performance of the proposed algorithm, various sizes of the assignment matrices with 10, 15, 20, 25, 30, 35 and 40 nodes are randomly generated and solved by using the G&N method and the presented approach. Moreover, three kinds of sample range, namely, (min = 10, max = 20), (min = 10, max = 40) and (min = 10, max = 80), are assigned to the values of cost and time. A simulated program is written in Turbo C and runs on a personal computer, and results are shown in Table 4, where upper means maximum cost in the matrix, and lower means minimum cost in the matrix. CPU is the CPU time (seconds) required to reach an optimal solution, n is the number of jobs or the size of the problem (n jobs x n workers), C and T are the cost and time of the assignment, Z is the sum of C and T, and b is the lower bound of T. It can be seen that, compared to G&N's method, the proposed approach optimizes the problems using much less CPU time in all tested examples, and the superiority becomes even obvious when the problem size increases. For instance, the proposed method requires almost only 13 % of the CPU time needed for G&N's method to reach the optimal solution for large size problems. This is believed to be a great saving. In addition, the proposed method generates a larger b value than that of G&N's method, this is advantageous because the feasible set of solutions can be significantly contracted, which leads to less computing time. Fig. 2 demonstrates the CPU time with the size of problems for the two methods.



(a) CPU time for sample range (10, 20)



(b) CPU time for sample range (10, 40)



(c) CPU time for sample range (10, 80)

Figure 2. CPU time for different sizes of problems

5. Conclusion

A matching technique in conjunction with the revised Hungarian algorithm to solve the bicriterion assignment problem is proposed in this paper. The presented approach converts the assignment problem into a bipartite graph and then repeatedly searches for the augmenting path with respect to the current matching. Unlike previous work done by G&N, the matching approach solves the problem at each stage with respect to the former stage, instead of solving the entire problem over again. This advantage significantly increases the computation efficiency. A Turbo C program is written to solve the different sizes of randomly generated problems. The performance of the matching technique is proved to be much better than G&N's method in all tested problems, the computation saving is found to increase with the size of problems. It is concluded that the proposed technique optimizes the bicriterion assignment problem efficiently and effectively. The limitation of the proposed method is that it may not guarantee an optimal solution due to its heuristic nature. Therefore, future research can be directed to the finding and verification of the global optimal solution.

References

Arsham, H. and Kahn, A. (1989). A simplex-type algorithm for general transportation problems: An alternative to stepping-stone. *Journal of the Operational Research Society*, Vol. 40, 581-590.

Barr, R.S., Glover, F., and Klingdam, D. (1977). The alternating basis algorithm for assignment problems, *Mathematical Programming*, 13, 1-13.

Bulut, H. (2016). Multiloop transportation simplex algorithm. *Optimization Methods and Software*, 32, 1-12.

Chen, S. G. (2015). Optimal double-resource assignment for a distributed multistate network. *Journal of Industrial & Management Optimization*, 11 (4), 1375-1391.

Das, U., Babu, A., Khan, A., Helal, T. U. (2014). Logical development of Vogel's approximation method (LD-VAM): An approach to find basic feasible solution of transportation problem. *International Journal of Scientific & Technology Research*, 3, 42-48.

Degroote, H., Velarde, J., and Causmaecker, P. (2018). Applying algorithm selection-a case study for the generalized assignment problem. *Electric Notes in Discrete Mathematics*, 69, 205-212.

Fisher, M. L., Jaikumar, R. and Vanm W. (1986). A multiplier adjustment method for the generalized assignment problem. *Management Science*, 32, 1095-1103.

Gabrovšek, B., Novak, T., Povh, J., Rupnik, P. D., Žerovnik, J. (2020). Multiple Hungarian method for *k*-assignment problem. *Mathematics*, 8 (11).

Garfinkel, R. S. (1971). an improved algorithm for bottleneck assignment problem. *Operations Research*, 18, 1717-1751.

Geetha, S., and Nair, K. P. K. (1993). A variation of the assignment problem. *European Journal of Operational Research*, 68, 422-426.

Hung, M. S. and Rom, W.O. (1980). Solving the assignment problem by relation. *Operations Research*, 28, 969-982.

Karsu, Ö. and Azizoğlu, M. (2012). the multi-resource agent bottleneck generalised assignment problem. *International Journal of Production Research*, 5 (2), 309- 324.

Khan, A., Rabbani, Q. and Quddoos, A. (2020). Modified Hungarian method for unbalanced assignment problem with multiple jobs. *Applied Mathematics and Computation*, 361, 493-498.

Mazzola, J. B. and Neebe, A. W. (1993). An algorithm for the bottleneck generalized assignment problem. *Computers and Operations Research*, 20, 355-362.

Mazzola, J. B. (1989). Generalized assignment with nonlinear capacity interaction. *Management Science*, 35, 923-941.

Mazzola J. B. and Neebe A. W. (1988). Bottleneck generalized assignment problems. *Engineering Costs and Production Economy*, 14, 61-65.

Moussavi, S. E., Mahdjoub, M., and Grunder, O. (2018). A hybrid heuristic algorithm for the sequencing generalized assignment problem in an assembly line. *IFAC- Papers Online*, 51 (2), 695-700.

Munapo, E. (2020). Development of an accelerating Hungarian method for assignment problems, *Eastern-European Journal of Enterprise Technologies*, 4, 6-13.

Palaniyappa, R. (2016). A study on northeast corner method in transportation problem and using of object oriented programming model. *Applied Mathematics*, 98, 42649-42641.

Posta, M., Ferland, J. A., and Michelon, P. (2012). An exact method with variable fixing for solving the generalized assignment problem, *Computational Optimization and Applications*, 52, 629-644.

Prasad, A. and Singh, D. (2020). Modified least cost method for solving transportation problem. *Engineering Sciences*, 2, 269-280.

Ross, G. T. and Soland, R. M. (1977). Modeling facility location problems as generalized problems, *Management Science*, 24, 345-357.

Shopov, V. K. and Markova, V. D. (2021). Application of Hungarian algorithm for assignment problem. *Proceedings of the 2021 International Conference on Information Technologies*. 1-4, IEEE.

Papadimitriou, C. H. and Steiglitz, K. (1982). Combinatorial Optimization. *Algorithms and Complexity*.

Wu, W., Iori, M., Martello, S. and Yagiura, M. (2018). Exact and heuristic algorithms for the interval min-max regret generalized assignment problem. *Computers and Industrial Engineering*, 125, 98-110.

Dr. Ding-Tsir Chang obtained Ph.D. in Industrial Engineering from Tsing Hua University, Taiwan. He is now an Associate Professor in the Department of Industrial Management at Chung Hua University, Taiwan. Dr. Chang's research areas include mathematical programming and decision making

Dr. Hsien-Hong Lin obtained Ph.D. in Higher Education Administration from Kent University, USA. He is now an Associate Professor in the Department of Logistics Engineering at Huaiyin Institute of Technology, China. Dr. Lin's research areas include logistics management and project management.

Dr. Su-Hui Chen obtained Ph.D. in Technology Management from Chung Hua University, Taiwan. She is now an Associate professor in the Department of Human Resources Management at Huaiyin Normal University, China. Dr. Chen's research areas include knowledge management and project management.

Prof. Chiu-Chi Wei obtained Ph.D. in Engineering Management from the University of Missouri, USA. He is now a professor in the Department of Industrial Management at Chung Hua University, Taiwan. Professor Wei has published 80 papers and authored eight books. He specializes in project management.