



# 풍력 발전기 평가를 위한 수집 행렬 데이터 절감 알고리즘 개발

이제현<sup>1)</sup> · 최정철<sup>2)\*</sup>

## Developing Novel Algorithms to Reduce the Data Requirements of the Capture Matrix for a Wind Turbine Certification

Jehyun Lee<sup>1)</sup> · Jungchul Choi<sup>2)\*</sup>

Received 23 September 2019 Revised 8 November 2019 Accepted 16 December 2019

**ABSTRACT** For mechanical load testing of wind turbines, capture matrix is constructed for various range of wind speeds according to the international standard IEC 61400-13. The conventional method wastes considerable amount of data by its invalid data policy -segment data into 10 minutes then remove invalid ones. Previously, we have suggested an alternative way to save the total amount of data to build a capture matrix, but the efficient selection of data has been still under question. The paper introduces optimization algorithms to construct capture matrix with less data. Heuristic algorithm (simple stacking and lowest frequency first), population method (particle swarm optimization) and Q-Learning accompanied with epsilon-greedy exploration are compared. All algorithms show better performance than the conventional way, where the distribution of enhancement was quite diverse. Among the algorithms, the best performance was achieved by heuristic method (lowest frequency first), and similarly by particle swarm optimization: Approximately 28% of data reduction in average and more than 40% in maximum. On the other hand, unexpectedly, the worst performance was achieved by Q-Learning, which was a promising candidate at the beginning. This study is helpful for not only wind turbine evaluation particularly the viewpoint of cost, but also understanding nature of wind speed data.

**Key words** Wind power(풍력), Wind turbine(풍력 발전기), Wind speed(풍속), Optimization(최적화), Artificial intelligence(인공지능), Algorithm(알고리즘)

### Nomenclature

$v_{in}$  : cut-in wind speed, m/s  
 $v_r$  : rated wind speed, m/s

### Subscript

SS : simple stacking  
PSO : particle swarm optimization

1) Senior Researcher, Platform Technology Laboratory, Korea Institute of Energy Research

2) Senior Researcher, Jeju Global Research Center, Korea Institute of Energy Research

\*Corresponding author: [jungchul.choi@kier.re.kr](mailto:jungchul.choi@kier.re.kr)  
Tel: +82-64-800-2227

### 1. 서론

새로운 모델의 풍력발전기를 출시하려면 형식 인증(type

certification)을 받아야 하고, 이를 위해 형식 시험(type testing)을 거쳐야 한다. 형식 시험의 필수 항목인 기계하중 시험은 기상장비가 구비된 곳에서 풍력발전기를 구동시키며 부품에 가해지는 하중을 계측하여 분석하는데 이를 위해 충분한 데이터 확보가 우선되어야 한다. 데이터 개수는 IEC 61400-13에 정의된 대로 풍속 구간별로 20개 또는 10개 이상이 되어야 한다. 이렇게 풍속별로 확보된 데이터 개수를 정리한 표를 수집 행렬(capture matrix)이라 한다<sup>[1]</sup>. 수집 행렬을 작성하기 위해 일반적으로는 수집한 데이터를 10분 단위로 분할한 후(이하 선분할 방식) 비유효 데이터를 삭제한다. 비유효 데이터는 풍향, 생산전력, 풍속 등의 기준으로 판별한다. 10분 데이터 중 1초라도 비유효 데이터가 있으면 삭제하기 때문에 데이터 손실 비중이 높다. 이에 따라 수집 행렬을 구성하기 위한 유효 데이터의 숫자를 확보하기 위해 총 수집 기간이 길어지는 문제가 발생하고 있다.

기계하중 시험을 위한 데이터 수집 기간이 길어짐에 따라 풍력 발전기 평가 비용이 상승하고, 이는 풍력 발전기 개발 및 제조 업체의 부담으로 작용한다.

이를 개선하고자 비유효 데이터를 먼저 걸러낸 후 나머지 데이터를 이용해 수집 행렬을 작성하는 방식을 제안한 바 있다(이하 후분할 방식)<sup>[2]</sup>. 위의 제안된 후분할 방식은 선분할 방식과 다른 방식을 제시하는 내용만을 담고 있으며 구체적인 방법은 제시하고 있지 않다. 본 연구에서는 후분할 방법들을 제시하여 수집 행렬 작성에 요구되는 데이터량을 줄이도록 한다. 이를 위해 경험적 학습법(heuristic), 입자군집법(particle swarm), 강화학습(Q-Learning)을 대상으로 수집 행렬 작성 성능을 비교하였다.

## 2. 방법론

### 2.1 선분할 방식

수집 데이터를 10분 단위로 분할 후 유효 데이터를 선별하는 선분할 방식은 Fig. 1(b)에 드러난 바와 같이 소수의 비유효 데이터가 있는 경우에도 10분 데이터 전체를 삭제하므로 유효 데이터를 충분히 활용하지 못하는 리스크가 있으나 데이터를 10분마다 시간순으로 분류하므로 데이터

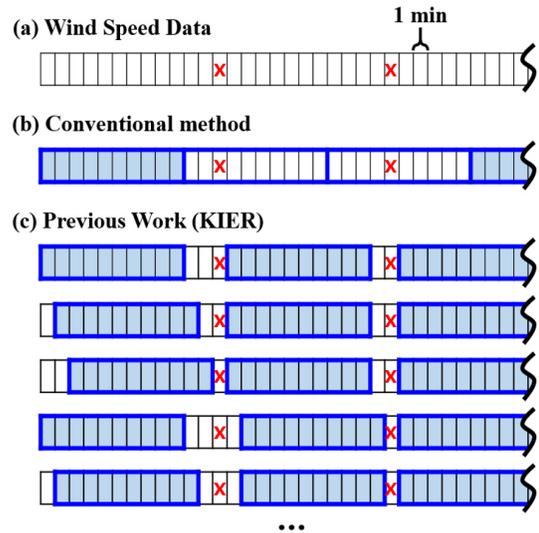


Fig. 1. Example of capture matrix construction with (a) given wind speed data, including invalid ones (x). Rules for data collection by (b) conventional method and (c) our previous suggestions are marked as blue boxes.

선택에 대한 고민은 필요치 않다. 반면 후분할 방식은 Fig. 1(c)와 같이, 선분할 방식에 비해 데이터를 효율적으로 사용할 수 있으나 수집 행렬 작성시 경우에 수가 많아 데이터 선택에 따른 효율성 편차가 발생한다. 모든 경우에 대한 전수조사(brute force)를 수행하여 최적의 경우를 선택하면 좋겠으나 경우의 수가  $> 10^{500}$ 개로 현실적으로 구현하기 어렵다. 데이터 하나당 선택 가능한 경우의 수가 10가지이며 수집 행렬 작성에는 약 500개의 데이터가 요구되기 때문이다. 따라서 데이터를 효율적으로 조합할 수 있는 알고리즘 선택이 매우 중요하다.

### 2.2 휴리스틱 알고리즘

휴리스틱 알고리즘은 경험과 사전 지식을 바탕으로 만든 알고리즘으로, 엄격한 수학적 증명을 토대로 하지는 않지만 경험적인 최적 조건을 구현한 알고리즘을 말한다. 본 연구에서는 (1) 선택 가능한 유효 데이터를 최대한 앞서부터 쌓아가는 단순 적재(Simple Stacking) 방식과 (2) 채워야 하는 수집 행렬 각 구간의 데이터 밀도를 비교하여 상대적인 밀도가 낮은 구간을 우선적으로 채우는 최소 빈도 방식을 적용하였다.

### 2.3 강화학습

강화학습은 방대한 레이블 데이터를 이용해 학습을 시키는 지도학습이나 레이블 없는 데이터간의 연관성을 파악하는 비지도학습과는 달리 시행착오를 반복하며 최선의 결과를 찾는 방법론이다. 강화학습의 여러 방법론 중 본 연구에서는 Q-Learning을 적용했다. 반복학습 당시까지 알려진 최적의 경로를 추종하는 특성상 국소 최적점(local optimum)에 빠지기 쉬운 단점이 있어 이를 해소하기 위해 임의의 동작을 하는 탐험(exploration) 과정이 필요하지만 학습이 많이 진행된 후반부에는 탐험의 비중이 줄어야 한다. 본 연구에서는 랜덤 선택 확률을 반복 수행 횟수에 반비례하여 지수함수적으로 줄여나가는 엡실론 그리디( $\epsilon$ -greedy) 방법을 적용하였다<sup>[3]</sup>.

### 2.4 개체군 기반 알고리즘

컴퓨터의 연산 능력을 바탕으로, 가능한 여러 조합을 동시에 테스트하는 방식이다. 본 연구에서는 입자군집(Particle Swarm) 최적화 기법을 사용하였다<sup>[4,5]</sup>. 입자 군집 방법론은 병렬 탐색을 이용해 국부 최적해에 쉽게 빠지지 않는 장점이 있어 범용적으로 사용되고 있으며, 다른 방법론들과 조합이 용이하여 혼종 방법론의 재료로도 많이 사용된다<sup>[6]</sup>. 연산 시간과 전역 최적점 탐색 능력을 고려한 사전 평가에서 입자 수는 50개, 최대 반복 수행은 100회로 설정하였으며, 이를 10회 반복하여 그 중 최적값을 채택하였다.

## 3. 수집 행렬 구축

### 3.1 풍속 데이터 분석

제주시 구좌읍 김녕리에 위치한 풍력 발전기에서 데이터를 수집하였으며, 풍력 발전기의 상세 제원은 Table 1과 같다.

Table 1. Wind turbine specification

Rated power	5,5 MW
Rated rotational speed	11,75 rpm
Rotor diameter	139 m
Hub height	100 m

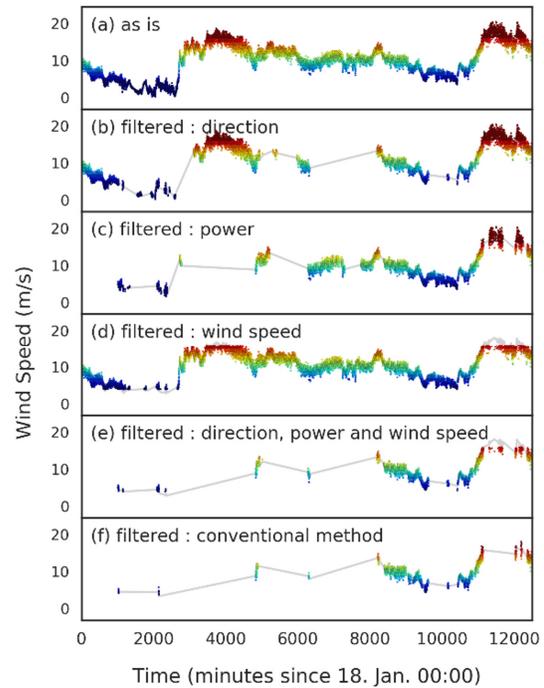


Fig. 2. Residual wind speed data after various data screenings. (a) As received, filtered only by (b) wind direction, (c) generated power and (d) wind speed. (e) Applying all together, compared with (f) by conventional method.

2019년 1월 18일 0시 0분부터 2월 18일 23시 50분까지 측정된 총 46021개의 풍속 데이터를 입력받아 일부를 Fig. 2(a)에 도시하였다. IEC 61400-13 규정에 따라 풍향이 허용범위를 벗어난 구간, 전력이 생산되지 않은 구간, 풍속이 시동풍속( $v_{in} = 4$  m/s) 미만이거나 중단풍속( $v = 16$  m/s) 이상 구간 제외 및 이를 동시에 적용한 결과를 Fig. 2(b~e)에 도시하고, 비교를 위해 선분할 방식 필터링 결과를 Fig. 2(f)에 나타내었다. 선분할 방식에서는 12,820개만 남은 데이터가 Fig. 2(e)의 후분할 방식에서는 14,564개 남아 선분할 방식 대비 14%(1,744개) 가량 많으므로(예: Fig. 2(e)와 (f)의 Time = 11000부근 비교), 본 데이터들이 효과적으로 활용될 경우 완료시간을 크게 앞당길 것이라 기대할 수 있다. 수집 행렬 작성 알고리즘에 따른 데이터 사용량을 비교하기 위해 각 날짜의 첫 데이터로부터 수집 행렬을 작성하고 그 결과를 비교하였다. 이 중 2월 1일 이후는 모든 방법론에서 유효 데이터 부족으로 수집 행렬 작성이 실패하였으며, 1월 20일 데이터는 전력 생산 기준을 만족하지 못하여 전체가 비유효 데이터로 판별되었다.

따라서 이들을 제외한 총 13일의 첫 데이터부터 수집 행

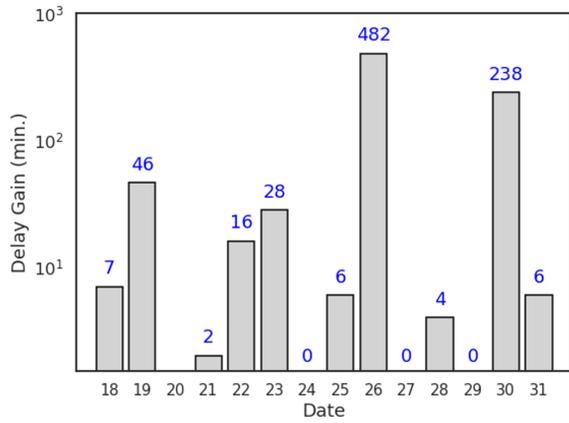


Fig. 3. Delay gain with respect to conventional method. Numbers of reduced data attached on top.

렬을 작성하고 그 결과를 이용해 알고리즘 성능을 평가하였다. 선분할 방식과 후분할 방식의 수집 행렬 작성 착수 시점을 비교하여 Fig. 3에 정리하였다. 후분할 방식 활용 시 평균 64분, 최대 482분 빠르게 수집 행렬 작성을 개시할 수 있음을 알 수 있다.

### 3.2 수집 행렬 구축 과정 비교

본 연구의 대상 터빈이 만족해야 하는 최소개수가 명시된 수집행렬을 Table 2에 도시하였다. IEC 61400-13에서 제시하는 수집 행렬 작성 규정에 따라 시동풍속부터 정격풍속( $v_R = 12 \text{ m/s}$ )보다 2 m/s 높은 14 m/s까지는 구간마다 10분 데이터를 20개 이상씩 수집하고 그 이후부터 종단풍속(16 m/s)까지는 10분 데이터를 10개 이상씩 수집해야 한다. 따라서 필요로 하는 10분 데이터 개수는 최소 220개로, 시간으로는 2,200분에 해당한다. 그러나 현실적으로 비유

Table 2. Capture matrix

Wind speed (m/s)	4~5	5~6	...	13 ~ 14	14 ~ 15	15 ~ 16
Counts	20	20	...	20	10	10

Table 3. Completed capture matrix, construction started from 18<sup>th</sup> Jan. 00:00

Wind speed (m/s)	4 ~ 5	5 ~ 6	6 ~ 7	7 ~ 8	8 ~ 9	9 ~ 10	10 ~ 11	11 ~ 12	12 ~ 13	13 ~ 14	14 ~ 15	15 ~ 16
Counts	20	55	60	101	95	99	85	32	45	65	21	17

Table 4. Initial and final time of capture matrix construction, with duration of conventional method

Initial		Final	Capture matrix construction duration (minutes)
Date	Time (actual)	Date and time	
01/18	17:00	02/08 16:10	20 days 57 hr, 22 min.
01/19	11:40	02/08 17:10	20 days 13 hr, 18 min.
01/21	08:30	02/08 23:30	18 days 37 hr, 12 min.
01/22	09:00	02/08 23:30	17 days 36 hr, 06 min.
01/23	17:00	02/08 23:30	16 days 16 hr, 06 min.
01/24	00:00	02/08 23:30	15 days 58 hr, 18 min.
01/25	01:10	02/08 23:30	14 days 55 hr, 20 min.
01/26	08:30	02/08 23:30	13 days 37 hr, 12 min.
01/27	00:00	02/08 23:30	12 days 58 hr, 18 min.
01/28	14:10	02/08 23:30	11 days 23 hr, 08 min.
01/29	00:00	02/08 23:30	10 days 58 hr, 18 min.
01/30	04:00	02/10 14:50	11 days 27 hr, 02 min.
01/31	00:40	02/10 18:50	10 days 45 hr, 10 min.
Average			15 days 1 hr, 27 min.

효 데이터가 많고 데이터가 불균일하게 분포하고 있어 비효율적인 선택이 불가피하기 때문에 실제로는 평균적으로 2배 이상, 약 500개 가량의 데이터를 수집해야 수집 행렬을 완결할 수 있다. 선분할 방식에 의해 1월 18일부터 완성된 수집 행렬을 Table 3에, 13일간의 수집행렬 작성 결과를 Table 4에 요약하였다. 목표값(Table 2)보다 훨씬 많은 15일분의 데이터가 소요됨을 알 수 있다.

#### 1) 단순 적재 방식

단순 적재 방식은 유효 데이터를 적재할 때, 선택 가능한 데이터 중 가장 먼저 등장하는 값을 적재하는 방식이다. 최대한 앞의 데이터를 반복해서 적재하면 최종적으로 수집 행렬을 조기에 작성 완료할 수 있으리라는 기대를 바탕으로 선택된 방법론이다. 1월 18일 데이터를 시작으로 수집 행렬을 작성하는 과정 중 첫 6개의 데이터를 수집하는 과정을 Table 5(a)에 도시하였다. 첫 1,012개의 데이터가 비유효 처리되어 Time index가 1013부터 시작하고 있으며 10분 평균 풍속을 적재하기 때문에 이상적으로는 Time index의 간격이 10이 되어야 하지만, 비유효 데이터로 인해 데이터 간격이 균일하지 않아 3번째와 4번째 데이터 간격은 950 (~16시간)에 이르기도 한다.

Table 5. An example of capture matrix construction by Heuristic methods, for 18<sup>th</sup> Jan

(a) Capture matrix under construction

Data No. (Seq.)	Time stamp	Time index (acc. min.)	Wind speed (m/s)	Capture matrix											
				20	20	20	20	20	20	20	20	20	20	10	10
1	2019-01-18 16:53:00	1013	5,171	0	1	0	0	0	0	0	0	0	0	0	0
2	2019-01-18 17:03:00	1023	4,680	1	1	0	0	0	0	0	0	0	0	0	0
3	2019-01-18 19:04:00	1144	4,089	2	1	0	0	0	0	0	0	0	0	0	0
4	2019-01-19 10:54:00	2094	4,546	3	1	0	0	0	0	0	0	0	0	0	0
5	2019-01-19 11:07:00	2107	4,587	4	1	0	0	0	0	0	0	0	0	0	0
6	2019-01-19 11:17:00	2117	4,305	5	1	0	0	0	0	0	0	0	0	0	0

(b) Candidates for next capture matrix element

Cand. Data No.	Time stamp	Time index (acc. min.)	Wind speed (m/s)	Additional capture matrix											
				20	20	20	20	20	20	20	20	20	20	10	10
-	2019-01-19 11:18:00	2118	4,268	0	0	0	0	0	0	0	0	0	0	0	0
-	2019-01-19 11:19:00	2119	4,176	0	0	0	0	0	0	0	0	0	0	0	0
-	2019-01-19 11:20:00	2120	4,140	0	0	0	0	0	0	0	0	0	0	0	0
-	2019-01-19 11:21:00	2121	4,117	0	0	0	0	0	0	0	0	0	0	0	0
-	2019-01-19 11:22:00	2122	4,017	0	0	0	0	0	0	0	0	0	0	0	0
-	2019-01-19 11:23:00	2123	4,076	0	0	0	0	0	0	0	0	0	0	0	0
-	2019-01-19 11:24:00	2124	4,156	0	0	0	0	0	0	0	0	0	0	0	0
1	2019-01-19 11:32:00	2132	5,047	0	+1	0	0	0	0	0	0	0	0	0	0
2	2019-01-19 11:33:00	2133	5,022	0	+1	0	0	0	0	0	0	0	0	0	0
3	2019-01-19 11:34:00	2134	4,998	+1	0	0	0	0	0	0	0	0	0	0	0
4	2019-01-19 11:35:00	2135	4,998	+1	0	0	0	0	0	0	0	0	0	0	0
5	2019-01-19 11:36:00	2136	4,961	+1	0	0	0	0	0	0	0	0	0	0	0
6	2019-01-19 11:37:00	2137	4,887	+1	0	0	0	0	0	0	0	0	0	0	0
7	2019-01-19 11:38:00	2138	4,754	+1	0	0	0	0	0	0	0	0	0	0	0
8	2019-01-19 11:39:00	2139	4,635	+1	0	0	0	0	0	0	0	0	0	0	0
9	2019-01-19 11:40:00	2140	4,518	+1	0	0	0	0	0	0	0	0	0	0	0
10	2019-01-19 11:41:00	2141	4,287	+1	0	0	0	0	0	0	0	0	0	0	0

본 예시에서 7번째 데이터로 선택될 수 있는 다음 후보군은 6번째 데이터의 10분 후인 Table 5(b)의 1월 19일 11시 32분부터 41분사이의 데이터이다. 단순 적재 방식은 이중 첫 번째 데이터를 선택하며, 후보군의 집합을 A라고 할 때 다음과 같이 표현된다.

$$\text{next data} = a_0 \text{ where } a_i \in A \forall i \tag{1}$$

단순 적재 방식으로 수집 행렬을 만들었을 때 시간 소모를 선분할 방식과 비교하여 Table 6에 정리하였다. 선분할

방식에 비해 데이터를 평균 2일 23시간, 최대 7일 21시간 덜 사용하였다.

2) 최소 빈도 방식

수집 행렬에서 원하는 데이터가 수집 가능한 데이터의 후보군의 맨 처음에 등장하지 않을 경우 수집 행렬이 완결되지 않는 단점이 있다. 이를 보완하기 위해 수집 행렬의 현재 미완 상태를 분석해서 가장 완성되지 못한 구간을 우선적으로 선택하는 방식을 최소 빈도 방식이라 한다. 수집 행렬의 어떤 구간 i의 목표값을  $CM_{t,i}$ , 비유효 데이터를 삭

**Table 6.** Initial and final time of capture matrix construction by Simple stacking, with time reduction w.r.t. IEC 61400-13

Initial		Final	Capture matrix construction time reduction (minutes)
Date	Time (actual)	Date and time	
01/18	16:53	02/01 07:24	7 days 21 hr, 12 min.
01/19	10:54	02/01 07:24	6 days 11 hr, 12 min.
01/21	08:28	02/02 12:55	4 days 20 hr, 15 min.
01/22	08:34	02/04 07:48	4 days 20 hr, 15 min.
01/23	16:22	02/04 07:48	4 days 20 hr, 15 min.
01/24	00:00	02/07 14:03	1 days 23 hr, 05 min.
01/25	01:04	02/07 14:03	1 days 23 hr, 05 min.
01/26	00:28	02/07 21:04	1 days 05 hr, 16 min.
01/27	00:00	02/07 21:54	1 days 03 hr, 14 min.
01/28	14:06	02/07 21:54	1 days 03 hr, 14 min.
01/29	00:00	02/07 22:34	1 days 01 hr, 22 min.
01/30	00:00	02/08 08:16	2 days 16 hr, 10 min.
01/31	00:34	02/08 08:46	2 days 24 hr, 18 min.
Average			2 day 22 hr, 41 min.

제한 유효 데이터 집합을  $D_v$ , 유효 데이터 집합 중 구간  $i$ 에 해당하는 부분집합을  $D_{v,i}$ 라고 하면 각 구간의 데이터 밀도는 아래와 같이 정의된다.

**Table 7.** Initial and final time of capture matrix construction by least frequency method, with time reduction w.r.t. conventional method

Initial		Final	Capture matrix construction time reduction (minutes)
Date	Time (actual)	Date and time	
01/18	16:53	01/31 15:10	8 days 02 hr, 12 min.
01/19	10:54	01/31 15:10	8 days 05 hr, 00 min.
01/21	08:28	02/02 13:03	6 days 26 hr, 03 min.
01/22	08:34	02/02 13:03	6 days 26 hr, 03 min.
01/23	16:22	02/02 13:03	6 days 26 hr, 03 min.
01/24	00:00	02/04 07:50	4 days 39 hr, 04 min.
01/25	01:04	02/04 07:50	4 days 39 hr, 04 min.
01/26	00:28	02/07 10:37	1 days 32 hr, 05 min.
01/27	00:00	02/07 11:56	1 days 28 hr, 22 min.
01/28	14:06	02/07 11:56	1 days 28 hr, 22 min.
01/29	00:00	02/07 21:05	1 days 06 hr, 01 min.
01/30	00:00	02/08 00:28	2 days 35 hr, 22 min.
01/31	00:34	02/08 00:28	2 days 45 hr, 22 min.
Average			4 days 6 hr, 50 min.

$$\rho_i = \frac{|D_{v,i}|}{CM_{t,i}} \quad (2)$$

각 구간의 데이터 밀도가 낮을수록 수집 행렬을 구성하기 어려운 구간이므로, 이를 보완하기 위해 데이터 밀도가 낮은 구간을 우선적으로 선택한다.

$$\text{next data} = \underset{i}{\text{minarg}}(\rho_i) \quad (3)$$

Table 5의 예시에 최소 빈도 방식을 적용하면, 수집 행렬의 2번째 구간(5~6 m/s,  $\rho_2 = 84.9$ )에 해당하는 후보들보다 1번째 구간(4~5 m/s,  $\rho_1 = 56.7$ )에 해당하는 후보를 우선적으로 선택하며, 이 중 가장 앞선 3번 후보를 다음 데이터로 확정한다. 최소 빈도 방식에서는 선분할 방식에 비해 평균적으로 4일 7시간 가량의 데이터를 덜 요구하였다.

### 3.2.2 강화학습

풍속 데이터 자체를 활용한 수집 행렬 최적화 알고리즘을 구현하고자 강화학습을 적용하였다. 본 연구에 적용한 Q-Learning은 특정 단계( $t$ )에서 채택한 데이터 선택을 상태( $S_t$ )로 정의하고, 이 단계에서 다음 데이터를 선택하는 행위를 동작( $A_t$ )이라 할 때,  $S_t$ 에서  $A_t$  수행에 따라 발생하는 효용의 기대값을  $Q(S_t, A_t)$ 라고 정의할 수 있다. 수집 행렬을 완성했을 때 주어지는 보상을  $R_t$ 라고 하면,  $Q$ 에 대한 방정식은 다음과 같이 주어진다<sup>[3]</sup>.

$$Q(S_t, A_t) \leftarrow (1 - \alpha) Q(S_t, A_t) + \alpha [R_t + \gamma \max_{A_{t+1}} Q(S_{t+1}, A_{t+1})] \quad (4)$$

본 연구에서 학습률(learning rate)  $\alpha = 0.5$ , 감쇠율(decaying rate)  $\gamma = 0.9$ 를 적용하였다. 보상은 단순 적재 방식에 비해 단축된 시간의 지수 함수로 정의하여 수집 행렬 작성 완료 시간을 단축했을 때 보상을 크게 인가하도록 설계하였다. 단순 적재 방식과 Q-Learning 방식의 수집 행렬 작성 완료 시간을 각기  $t_{f,ss}$ ,  $t_{f,ql}$ 라고 할 때 보상은 다음과 같다.

$$R_t = \exp(t_{f,ss} - t_{f,ql}) \quad (5)$$

Q-Learning은 매 순간의 보상이 높은 선택지를 따라가는 방식이기 때문에 국부 최소해(local minima)에 쉽게 빠지는 단점이 있다. 이를 해소하기 위해 기존의 경로를 최적화하는 탐색(exploitation)에 새로운 경로를 개척하는 탐험(exploration)을 특정 확률( $\epsilon$ )로 수행하되 확률이 점차 감소하는 decayed  $\epsilon$ -greedy를 적용했다.  $\epsilon$ 는 반복 횟수에 반비례하여 1,000회 반복수행 후에는 0.1%로 탐험하도록 설정하였다.

Q-Learning을 통한 학습과정을 Fig. 4에 도시하였다. 수집 행렬 완성 시각 단축에 보상을 주었으므로 반복수행에 따라 수집 행렬 완성 시점(a)과 수집 행렬에 소요된 데이터 수(b)는 우하향을 그릴 것으로 예상되고, 반면 Q 행렬 점유율(c)은 학습을 시작할 때는 아무런 정보가 없지만 학습이 진행되면서 점차적으로 채워지는 우상향을 보일 것으로 예상되며, 실제로 전체적으로 이러한 거동을 보이고 있다. 그러나 Fig. 4(a)를 보면 반복수행에 따라 최적점을 찾아가는 모습이 잡음에 가려 잘 드러나지 않는다.

최대 30,000회까지 진행된 반복수행 이후에도 Q-Learning이 수렴하지 못하고 계속 진동하는 모습이 보이는데, 반복수행 최대 횟수를 훨씬 많이 늘려도 마찬가지로 양상을 보인다. 비슷한 수준의 해를 여러 다른 경로를 통해 얻을 수 있고, 이들의 경로가 서로 크게 얽혀 있음으로 인해 발생하는 현상이다. 반복수행이 많이 진행되어 탐험 확

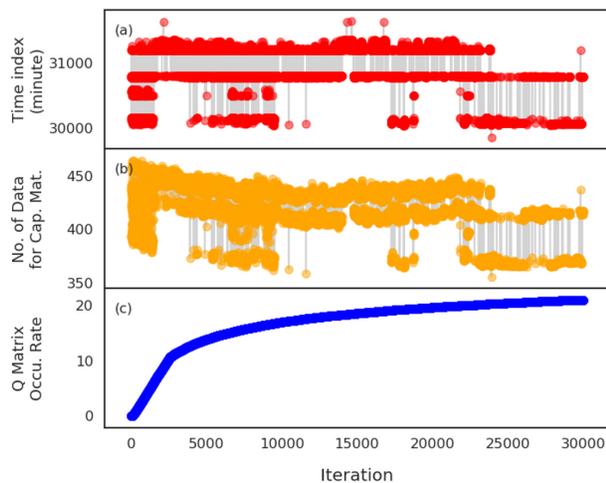


Fig. 4. Process of Q-Learning, starting from 26<sup>th</sup> Jan. (a) Time index when capture matrix construction is finalized, (b) number of total data for capture matrix construction, and (c) occupation rate of  $Q(s, a)$  matrix.

률이 0에 가깝게 낮아져 있다 하더라도 Q 행렬이 많이 채워져 있기 때문에, 탐험으로 인해 발생한 국부 최적해로부터 전파되어 채워진 Q 행렬로 인해 전역 최적해로부터 멀어지는 결과를 가져온다.

이는 Q-Learning의 본질적 한계로, 마르코프 상태(Markov state)를 전제 조건으로 가지기 때문이다<sup>[3,7]</sup>.  $s_{t+1}$ 는 오직  $s_t$ 에만 영향을 받으며 그 이전의 상태로부터는 영향을 받지 않는다는 의미로, 현재는 과거와 무관하다는 말로 정리할 수 있다. 그러나 1분 단위로 수집된 시계열 데이터를 10분 단위로 묶어 선택하는 수집 행렬 작성 과정의 특성상 과거 데이터에 대한 의존성이 매우 높기 때문에 국부 최소해에 의한 지속적인 학습 방해는 불가피한 현상으로 판단된다.

Q-Learning에 의한 수집 행렬 작성시 선분할 방식 대비 평균 2일 11시간, 최대 6일 55시간 가량의 데이터 절감 효과가 확인되었다(Table 8).

### 3.2.3 개체군 기반 알고리즘

입자 군집 방법론은 새나 곤충 무리의 사회적 행동에서 영감을 얻어 주어진 공간을 병렬적으로 탐색하기 위한 기법이다<sup>[4,5]</sup>. 공간에 분포하는 입자들은 각자의 속도와 방향을

Table 8. Initial and final time of capture matrix construction by Q-Learning, with time reduction w.r.t. conventional method

Initial		Final		Capture matrix construction time reduction (minutes)
Date	Time (actual)	Date and time		
01/18	16:53	02/01 18:00	6 days 55 hr, 10 min,	
01/19	10:54	02/02 13:31	6 days 09 hr, 03 min,	
01/21	08:28	02/04 18:58	4 days 11 hr, 08 min,	
01/22	08:34	02/07 06:13	1 days 43 hr, 05 min,	
01/23	16:22	02/07 06:12	1 days 43 hr, 06 min,	
01/24	00:00	02/07 11:48	1 days 29 hr, 06 min,	
01/25	01:04	02/07 11:57	1 days 28 hr, 21 min,	
01/26	00:28	02/07 17:43	1 days 14 hr, 11 min,	
01/27	00:00	02/07 21:07	1 days 05 hr, 23 min,	
01/28	14:06	02/07 21:30	1 days 05 hr, 00 min,	
01/29	00:00	02/07 22:47	1 days 01 hr, 19 min,	
01/30	00:00	02/08 12:02	2 days 07 hr, 00 min,	
01/31	00:34	02/08 12:52	2 days 14 hr, 22 min,	
Average				2 day 11 hr, 59 min,

가지고 공간을 훑으며 각자 획득한 개별 최적해(particle optimum)의 정보를 종합하여 전역 최적해(global optimum)를 찾는 방식이다. 이를 위해 각 반복마다 입자들이 알고 있는 개별 최소해의 위치를 교환하고 이들 중 가장 낮은 값을 가진 전역 최소해를 선정하여 사전에 정의된 가중치를 개별 최소해 방향, 전역 최소해 방향, 기존 단계 속도 벡터를 합산하여 새로운 공간을 탐색한다.

본 연구에서 수집 행렬의 첫 데이터를 포함하여 수집 행렬이 완결되기까지의 데이터 선택으로 구성된 벡터를 하나의 입자로 정의하였다. 수집 행렬 작성 과정을 모두 담을 수 있도록 최대 차원  $N$ 은 충분히 큰 값인 1,000으로 설정하였다. 수집 행렬 작성이  $n < N$  단계에서 완료되면 이후의 데이터는 nan 값으로 처리했으며,  $t$  단계에서의  $i$  번째 입자 벡터는 다음과 같이 표현된다.

$$a_{i,t} = \{A_{i,0}, A_{i,1}, A_{i,2}, \dots, A_{i,n}, \dots, A_{i,N}\} \quad (6)$$

입자 벡터는  $N$  단계에서 Q-Learning에서 수집 행렬이 완성된 뒤 행동들의 집합과 동일하나, Q-Learning은 수집 행렬 반복 작성시 행렬에 저장된 Q 값을 기반으로 처음부터 수행하는데 반해 입자 군집은 현재의 위치를 바탕으로 주변 공간을 탐색한다. 단순 적재 방식으로 작성한 수집 행렬을 입자 벡터의 초기값으로 설정하고, 초기 벡터를 임의의 값으로 설정하여 입자들이 다른 방향으로 움직이도록 하였다.

Fig. 5에 50개의 입자를 이용한 수집 행렬 최적화 과정

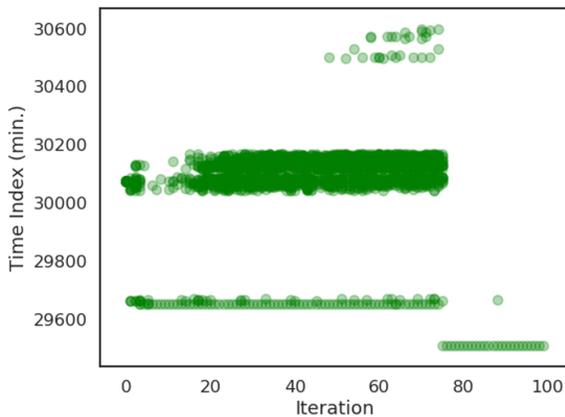


Fig. 5. Particle swarm optimization process of 50 particles with 100 iteration

( $t \leq 100$ )을 도시하였다. 초기값에서부터 단계별로 더 낮은 국부 최소해를 찾아가는 한편 일부 입자들은 다른 공간을 탐색하는 모습을 볼 수 있다. 그래프에 도시된 입자들이 전체 입자 수에 비해 적은데, 많은 입자들이 수집 행렬 작성에 실패하여 그래프에 표시되지 못하였다.

벡터 공간을 탐색하는 입자의 수와 반복 횟수가 많을수록 최소값 탐색이 유리하나 계산 시간 및 국소 최소해 문제를 최소화하기 위해 입자 군집 최적화를 반복하여 적용하는 방식을 채택하였다. 주어진 입자의 수와 반복 횟수에서 시스템 전체의 전역 최소값을 찾을 확률을  $p_g$ , 입자 군집 최적화 횟수(repeat)를  $m$ 이라고 할 때 전역 최소값을 찾을 확률  $P = 1 - (1 - p_g)^m$ 이다. 본 연구에 적용했듯  $m = 10$ 일

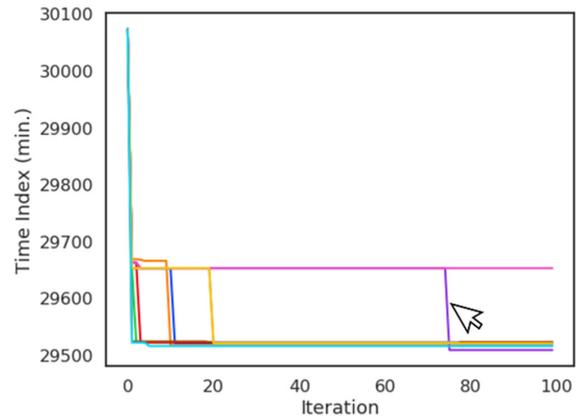


Fig. 6. Particle swarm optimization processes of 10 repeats. The colors of the line stands for trajectory of each repeat. The optimization process shown on Fig. 5 is marked with an arrow.

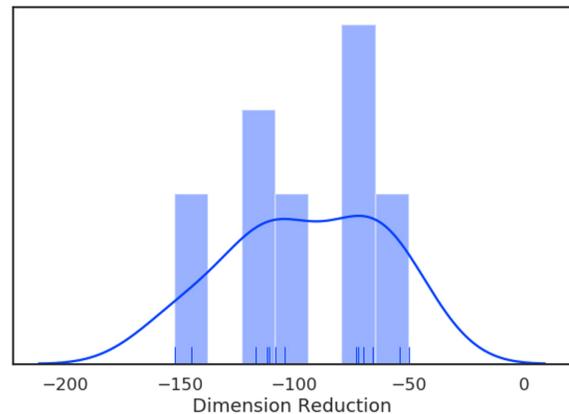


Fig. 7. Distribution of dimension reduction, during particle swarm optimization process. The bar presents histogram with kernel density estimation (solid line), as the ticks denote each data

때  $p_g = 0.3$ 일 경우  $P = 0.97$ ,  $p_g = 0.1$ 에 불과하더라도  $P = 0.65$ 가 되어 전역 최소값 탐색 능력이 크게 상승한다.

Fig. 6에 도시한 입자군집 최적화 10회 반복 과정에서 반복 수행의 이점을 확인할 수 있다. 화살표로 표시한 Fig. 5의 사례를 포함하여 많은 경우 Time index가 29520 부근일 때를 최적해로 판단하고 있으나 적은 수가 29512를 달성하였다. 한편, 단순 적재 방식에 비해 입자 군집 최적화 적용시 수집 행렬 작성 데이터량이 크게 감소한다. 수집 행렬 최적화 후, 10분 단위 데이터 개수 기준으로 선분할 방식과 비교하여 평균적으로 100개 가량, 최대 약 152개(1월 26일)까지 줄어들었음을 확인하였다(Fig. 7).

### 3.3 수집 행렬 구축 결과 비교

본 논문에서 평가한 방법론 중 최적의 수집 행렬 작성 알고리즘을 선정하기 위해, 각 방법론에서 수집 행렬 작성을 위해 요구된 데이터 양을 Fig. 8에 나타내었다. 이 때 데이터 요구량은 비효율 데이터를 포함한 각 날짜의 첫 데이터로부터 수집 행렬이 완성된 시점까지의 데이터를 기준으로 산정하였으며, 날짜별 요구 데이터 분포를 바이올린 플롯으로 도시하고 개별 데이터를 점으로 표기하였다.

Table 9. Initial and final time of capture matrix construction by particle swarm, with time reduction w.r.t. conventional method

Initial		Final	Capture matrix construction time reduction (minutes)
Date	Time (actual)	Date and time	
01/18	16:53	01/31 15:23	8 days 01 hr, 23 min,
01/19	10:54	02/01 07:34	7 days 24 hr, 00 min,
01/21	08:28	02/02 13:32	6 days 24 hr, 22 min,
01/22	08:34	02/04 07:53	4 days 39 hr, 01 min,
01/23	16:22	02/04 07:55	4 days 38 hr, 23 min,
01/24	00:00	02/07 10:18	1 days 33 hr, 00 min,
01/25	01:04	02/07 10:17	1 days 33 hr, 01 min,
01/26	00:28	02/07 11:56	1 days 28 hr, 22 min,
01/27	00:00	02/07 14:25	1 days 22 hr, 17 min,
01/28	14:06	02/07 20:44	1 days 06 hr, 22 min,
01/29	00:00	02/07 21:27	1 days 05 hr, 03 min,
01/30	00:00	02/07 23:54	2 days 37 hr, 08 min,
01/31	00:34	02/08 02:05	2 days 41 hr, 21 min,
Average			3 days 10 hr, 24 min,

데이터를 선분할하는 방식은 수집 행렬 작성 착수일에 따른 요구 데이터량의 편차가 심한데 반해 비효율 데이터를 먼저 삭제하는 후분할 방법론들은 편차가 상대적으로 작고, 분포 또한 상대적으로 평균값 부근에 집중되어 있을 알 수 있다. 선분할 방식보다 평균적으로 적게는 1,483분(1일 1시간, Q-Learning, 1월 28일)부터 많게는 11,640분 이

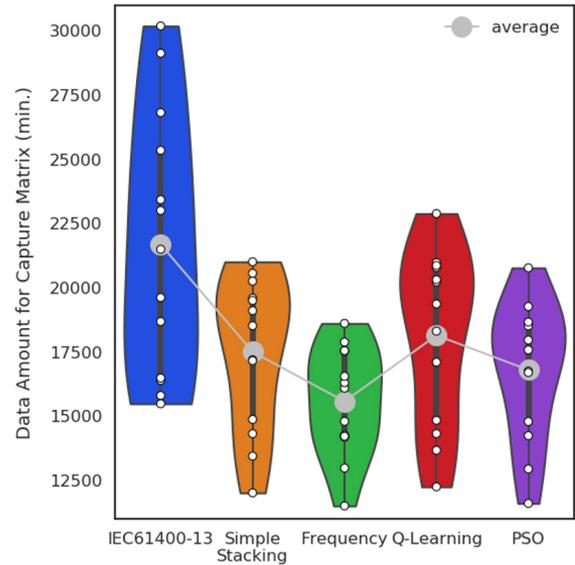


Fig. 8. Distribution of data amount required for capture matrix completion. The averages of the data are marked as large gray dots.

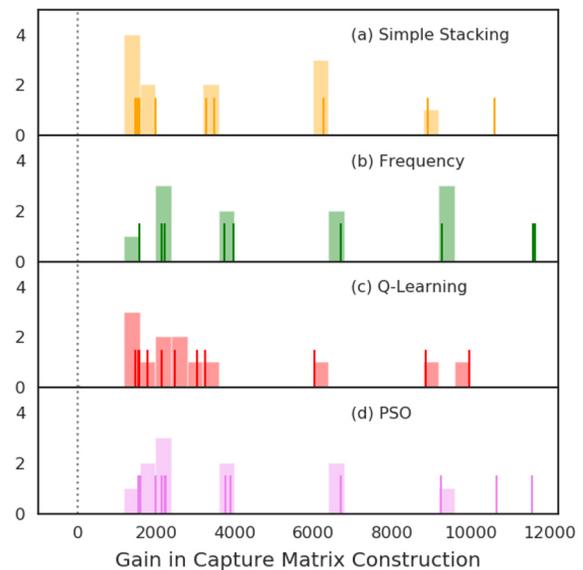


Fig. 9. Gain of each algorithms w.r.t. the conventional method. The border line at gain = 0 is marked with dotted line. The bar and ticks denote histogram and each data, respectively.

상(8일 2시간, 최소 빈도, 1월 18일) 후분할 방식 적용시 후분할 방식의 효율성 증대를 확인하였다. Q-Learning은 단순 적재 방식(2,241분, 2일 23시간 단축)보다 오히려 효율이 좋지 않았지만 Q-Learning 학습의 출발점이 단순 적재 방식으로 작성한 수집 행렬이므로 학습으로 작성한 수집 행렬이 초기값보다 좋지 않은 경우, 초기값을 사용하는 방식으로 성능 향상이 가능하다. 가장 좋은 성능을 보인 최소 빈도 방법론 적용시 선분할 방식 대비 평균적으로 28%(6,170분)의 데이터를 절감할 수 있었으며 특히 1월 19일 데이터로 작성하기 시작한 수집 행렬에서는 각기 최대 40%(11,640분)의 데이터 절감 효과가 확인되었다. 알고리즘에 따른 수집 행렬 데이터 절감 효과를 Fig. 9에 정리하였다. 여기서 주목할 또 하나의 지점은 선분할 방식 대비 성능열화 리스크이다.

단순 적재를 제외한 모든 방법론들이 확률적으로 최적점을 찾는 방식이기 때문에 여러 시도에도 불구하고 선분할 방식보다 좋은 해를 찾지 못할 가능성이 항상 존재하기 때문이다. 그러나 Fig. 9(a)를 보면 선분할 방식보다 성능이 열화된 경우가 모든 경우에서 존재하지 않아 후분할 방식 자체가 선분할 방식에 비해 데이터 효율성 측면에서 우월함을 확인하였다.

그러나 동일한 방법론 안에서도 수집 행렬 작성을 시작한 날짜에 따라 요구되는 데이터량의 최대-최소 차이가 평균값의 68%(선분할방식)에서 46%(최소 빈도 방법론)까지 달라지는 만큼 데이터의 시간에 따른 풍속 분포 양상에 따라 유리한 수집 정책이 바뀔 수 있어 알고리즘별 성능이 변동될 수 있는 여지가 있다.

#### 4. 결론

풍력 발전기 평가를 위한 수집 행렬 작성을 최적화하고자 단순 적재, 최소 빈도, Q-Learning, 입자 군집 방법을 적용하였다. 총 32일간의 데이터를 이용하여 13개의 시작 시간을 설정한 결과, 최소 빈도 방법론에서 최대 40%의 데이터 절감 효과 및 모든 경우에서 선분할 방식보다 나은 성능을 보였다. 위 알고리즘들은 딥러닝과 같이 고가의 계산 자원을 요구하지 않으면서 요구 데이터를 효과적으로 절감

할 수 있으므로, 풍력 발전기 평가에 즉시 적용하여 평가 일정 단축에 기여할 수 있을 것으로 판단된다.

본 연구는 평균 풍속이 높은 겨울철 데이터를 대상으로 실시하였으며, 고풍속 구간 데이터가 부족한 여름철 데이터를 이용할 경우 본 연구의 성능을 보다 정확히 파악할 것으로 예상된다. 다만 더 많은 데이터를 이용한 연구를 통하여 본 연구의 우수성을 확인할 필요가 있다.

#### 감사의 글

본 연구는 산업통상자원부의 “국산 수지를 적용한 해상용 풍력발전기 블레이드 개발 및 실증(2017-3010024850)” 및 “복수 풍력 단지 연계형 통합 제어 기술 개발 및 실증(2018-3010025170)” 과제로부터 지원을 받아 수행되었습니다.

#### References

- [1] International Electro-technical Commission, 2015, “IEC 61400-13: Wind turbines - Part 13: Measurement of mechanical loads”, International Electro-technical Commission (IEC), Geneva.
- [2] Choi, J., Lee, J., and Kang, S., 2019, “Method of optimizing capture matrix”, KR Patent No. 1020190063688.
- [3] Sutton, R.S., and Barto, A.G., 2015, “Reinforcement learning: an introduction”, 2nd ed., London, England: The MIT Press.
- [4] Kennedy, J., and Eberhart, R., 1995, “Particle swarm optimization”, Proceedings of ICNN’95-International Conference on Neural Networks, 4, 1942-1948.
- [5] Esmine, A.A.A., Coelho, R.A. and Matwin, S., 2015 “A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data”, Artif. Intell. Rev., 44(1), 23-45.
- [6] Peng, F., Tang, K., Chen, G., and Yao, X., 2010, “Population-based algorithm portfolios for numerical optimization”, IEEE Trans. Evol. Comput., 14(5), 782-800.
- [7] Gagnic, P.A., 2017, “Markov chains : from theory to implementation and experimentation”, Wiley.