

<https://doi.org/10.7236/JIIBC.2022.22.4.87>
JIIBC 2022-4-13

퍼펙트 9 다트 조합의 나눗셈 분기한정 알고리즘

Branch-and-Bound Algorithm for Division of Perfect Nine Dart Combinations

이상운*

Sang-Un Lee*

요약 본 논문은 501 다트게임의 최소 다트 총 투척 횟수인 9회에 대한 획득점수 조합을 찾는 연구를 수행하였다. 다트게임에서 한 번 투척으로 획득할 수 있는 최대 점수는 60점으로 $60 \times 8 + 21 \times 1 = 501$ 에 의거 60점을 8회, 21점을 1회 획득하면 9회 투척으로 퍼펙트 다트게임을 종료할 수 있다. 이를 9-다트 종료라 한다. 이와 같이 9회 투척으로 501점을 획득 가능한 점수의 조합에 관한 연구는 18개와 14개만 알려져 있으며, 전수탐색 알고리즘을 적용한 연구는 수행되지 않고 있다. 본 논문은 전형적인 전수탐색법인 분기한정법의 $O(2^n)$ 지수시간 수행 복잡도를 다항시간 복잡도로 단순화시키는 방법으로 나눗셈 분기한정 알고리즘을 제안하였다. 제안된 방법은 레벨을 8로 한정시키고, 501/60의 몫 레벨로 점프하고 역추적으로 이전 단계에서의 가능한 점수 조합만을 탐색하는 방법을 적용하였다. 제안된 알고리즘으로 찾은 9회의 퍼펙트게임의 가능한 점수 조합은 90개(101개 경우)를 얻었다.

Abstract This paper researched a study to find a combination of acquisition scores for 9 dart throws, which is the minimum number of dart tactile throws in 501 point dart games. The maximum score that can be obtained by throwing once in a dart game is 60 points, which can end the perfect dart game with 60 points eight times according to $60 \times 8 + 21 \times 1 = 501$, and if you earn 21 points once, you can finish the game with 9 throws. This is called 9-dart finish. As such, only 18 and 14 studies on the combination of scores that can obtain 501 points with 9 throws are known, and no studies have been conducted applying the exhaustive search algorithm. This paper proposed a division branch-and-bound algorithm as a method of simplifying the $O(2^n)$ exponential time performance complexity of the typical branch-and-bound method of a exhaustive search method, to polynomial time complexity. The proposed method limited the level to 8, jumped to a quotient level of 501/60, and backtracked to explore only possible score combinations in the previous level. The possible score combinations of the nine perfect games found with the proposed algorithm were 90(101 cases).

Key Words : 501 9-dart finish, Perfect combinations, Division B&B, Division jump, Backtracking

*정회원, 강릉원주대학교 과학기술대학 멀티미디어공학과
접수일자 2022년 2월 8일, 수정완료 2022년 7월 10일
게재확정일자 2022년 8월 5일

Received: 8 February, 2022 / Revised: 10 July, 2022 /
Accepted: 5 August, 2022

*Corresponding Author: sulee@gwnu.ac.kr
Dept. of Multimedia Eng., Gangneung-Wonju National
University, Korea

I. 서 론

다트 게임(dart game)은 다트 축으로 다트 판(dart board)의 점수 과녁을 명중하여 설정된 목표 점수를 가장 먼저 획득한 사람이 승자가 되는 실내 스포츠게임이다. 게임 규칙은 한 사람이 다트 축을 3회 던지는 라운드를 선수들이 번갈아 진행하여 8라운드 총 24회를 던지는 게임으로 목표 점수는 총점 301/501/701/ 901 중에서 501을 주로 적용하고 있다.

게임종류는 점수 합산(count up), 제로윈(X01), 승자 진출 식(knock out)과 표준 크리켓(standard cricket)이 있으며, 일반적으로 X01을 채택한다. X01은 획득한 점수를 501점에서 빼서 먼저 '0'을 만드는 사람이 승리하는 경기 규칙이다. 만약 남아 있는 점수보다 높은 점수를 득점하면 버스트(bust)로 점수가 던지기 전으로 되돌려진다.

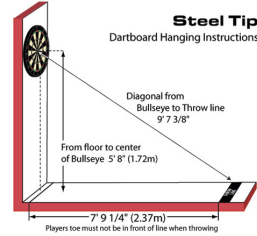
501 다트 게임에서 퍼펙트게임(perfect game)이란 501점을 가장 먼저 획득할 수 있는 최소 다트 축 투척 횟수이다. 다트 판에는 1,2,...,20의 20개 기본 점수와 2배(double, D), 3배(triple, T)의 점수, 과녁 중심 흑점인 내부 흑점(inner bullseye)의 50점, 흑점 가장자리 원인 외부 흑점(outer bullseye)의 25점으로 구성되어 있어 한 번에 획득할 수 있는 최고 점수는 60점(tripple 20, T20)이다. 따라서 $\frac{501}{60} = 8.35$ 로 최소 9회인 $60 \times 8 + 21 \times 1 = 501$ 로 T20 과녁에 8회, T7 과녁에 1회 맞추어야 501점에 가장 먼저 도달하는 퍼펙트게임의 승자가 된다. 따라서 이를 9-다트 경기 종료(9-dart finish)라 한다.^[1-11]

여기서 제기되는 문제는 이와 같이 "501점을 9회 만에 던져서 퍼펙트게임을 달성하는 경우 수는 $T20 \times 8 + T7 \times 1$ 이외에 몇 가지나 더 존재하는가?" 이다. 이에 대한 연구 결과는 수학적 공식 또는 알고리즘으로 증명된 사례가 전혀 없으며, 단지 18가지^[1-10], 또는 27가지^[11] 경우 수만이 제시되고 있다.

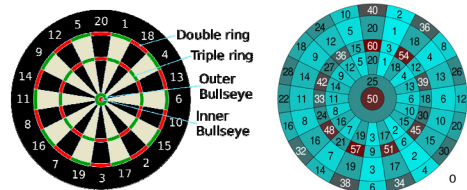
본 논문에서는 9-다트 경기 종료 문제(9-dart finish problem, 9-DFP)의 501점을 9회 투척으로 획득할 수 있는 가능한 모든 경우 수인 조합을 찾는 알고리즘을 제안한다. 2장에서는 다트게임 방법과 9-DFP의 501점 획득 경우 수와 관련된 연구 결과를 고찰한다. 3장에서는 나눗셈 분기한정 알고리즘(division branch-and-bound algorithm, DBBA)을 제안하고, 4장에서는 501점에 대해 DBBA로 가능한 조합 경우 수를 구한 결과를 제시한다.

II. 다트 게임 방법과 501 9-다트 게임 종료

경우 수 관련연구



(a) Distance of the Throw line - Steel Tip



(b) Dart board & Scores for each region

Score	Cases			Frequency	
60	-	-	T20	60(T20)	1
57	-	-	T19	57(T19)	1
54	-	-	T18	54(T18)	1
51	-	-	T17	51(T17)	1
50	IB	-	-	50(IB)	1
48	-	-	T16	48(T16)	1
45	-	-	T15	45(T15)	1
42	-	-	T14	42(T14)	1
40	-	D20	-	40(D20)	1
39	-	-	T13	39(T13)	1
38	-	D19	-	38(D19)	1
36	-	D18	T12	36(D18, T12)	2
34	-	D17	-	34(D17)	1
33	-	-	T11	33(T11)	1
32	-	D16	-	32(D16)	1
30	-	D15	T10	30(D15, T10)	2
28	-	D14	-	28(D14)	1
27	-	-	T9	27(T9)	1
26	-	D13	-	26(D13)	1
25	OB	-	-	25(OB)	1
24	-	D12	T8	24(D12, T8)	2
22	-	D11	-	22(D11)	1
21	-	-	T7	21(T7)	1
20	S20	D10	-	20(S20, D10)	2
19	S19	-	-	19(S19)	1
18	S18	D9	T6	18(S18, D9, T6)	3
17	S17	-	-	17(S17)	1
16	S16	D8	-	16(S16, D8)	2
15	S15	-	T5	15(S15, T5)	2
14	S14	D7	-	14(S14, D7)	2
13	S13	-	-	13(S13)	1
12	S12	D6	T4	12(S12, D6, T4)	3
11	S11	-	-	11(S11)	1
10	S10	D5	-	10(S10, D5)	2
9	S9	-	T3	9(S9, T3)	2
8	S8	D4	-	8(S8, D4)	2
7	S7	-	-	7(S7)	1
6	S6	D3	T2	6(S6, D3, T2)	3
5	S5	-	-	5(S5)	1
4	S4	D2	-	4(S4, D2)	2
3	S3	-	T1	3(S3, T1)	2
2	S2	D1	-	2(S2, D1)	2
1	S1	-	-	1(S1)	1

(c) Summary of scores

그림 1. 다트 게임과 점수
Fig. 1. Dart game & Scores

다트 게임은 그림 1의 (a)와 같이 2.37m 거리에서 원형 다트 판(dart board)에 다트 축을 던져 점수를 획득하는 방법이다.^[2,10]

(b)에는 과녁이 꽃힌 지점의 다트 판 획득 점수를 표기하고 있다. 이를 획득할 수 있는 모든 점수 내림차순으로 요약 정리하면 (c)와 같다.

501점을 9회 만에 던져서 퍼펙트게임을 달성하는 경우 수인 조합(combination)에 관한 연구를 요약하여 그

림 2에 제시하였다. (a)는 18가지^[1-10], (b)는 27가지^[11]를 제시하고 있음을 알 수 있다. (a)에 대해 대부분의 연구 결과는 3944 가지^[1,2,6,8,9], 혹은 4,944 가지^[7]를 제시하고 있으며 실제로는 18가지를 얻을 수 있다. (b)에서는 체크아웃 표(checkout table)의 서로 다른 경우 수를 고려하지 않은 경우 71가지를 원문에서 제시하였으나 3-3-3의 3 라운드 중 서로 다른 라운드에서 획득한 점수가 동일한 중복되는 경우를 제거하면 27가지를 얻을 수

Cases	#1	#2	#3	#4	#5	#6	#7	#8	#9	Scores	$s_j x_j$	
1	60	60	60	60	60	60	60	57	24	501	60x7+57x1+24x1	T20x7+T19x1+D12x1
2	60	60	60	60	60	60	60	51	30	501	60x7+51x1+30x1	T20x7+T17x1+D15x1
3	60	60	60	60	60	60	60	45	36	501	60x7+45x1+36x1	T20x7+T15x1+D18x1
4	60	60	60	60	60	60	57	54	30	501	60x6+57x1+54x1+30x1	T20x6+T19x1+T18x1+D15x1
5	60	60	60	60	60	60	57	50	34	501	60x6+57x1+50x1+34x1	T20x6+T19x1+IBx1+D17x1
6	60	60	60	60	60	60	57	48	36	501	60x6+57x1+48x1+36x1	T20x6+T19x1+T16x1+D18x1
7	60	60	60	60	60	60	54	51	36	501	60x6+54x1+51x1+36x1	T20x6+T18x1+T17x1+D18x1
8	60	60	60	60	60	60	51	50	40	501	60x6+51x1+50x1+40x1	T20x6+T17x1+IBx1+D20x1
9	60	60	60	60	60	57	57	57	30	501	60x5+57x3+30x1	T20x5+T19x3+D15x1
10	60	60	60	60	60	57	57	51	36	501	60x5+57x2+51x1+36x1	T20x5+T19x2+T17x1+D18x1
11	60	60	60	60	60	57	54	54	36	501	60x5+57x1+54x2+36x1	T20x5+T19x1+T18x2+D18x1
12	60	60	60	60	60	57	54	50	40	501	60x5+57x1+54x1+50x1+40x1	T20x5+T19x1+T18x1+IBx1+D20x1
13	60	60	60	60	60	51	50	50	50	501	60x5+51x1+50x3	T20x5+T17x1+IBx3
14	60	60	60	60	57	57	57	54	36	501	60x4+57x3+54x1+36x1	T20x4+T19x3+T18x1+D18x1
15	60	60	60	60	57	57	57	50	40	501	60x4+57x3+50x1+40x1	T20x4+T19x3+IBx1+D20x1
16	60	60	60	60	57	54	50	50	50	501	60x4+57x1+54x1+50x3	T20x4+T19x1+T18x1+IBx3
17	60	60	60	57	57	57	57	57	36	501	60x3+57x5+36x1	T20x3+T19x5+D18x1
18	60	60	60	57	57	57	50	50	50	501	60x3+57x3+50x3	T20x3+T19x3+IBx3

(a) 18-combinations for 501 9-dart finish^[1,2,6,7,8,9]

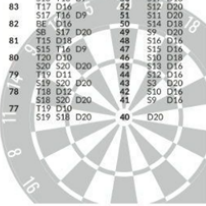
CHECKOUT TABLE

3 DART FINISHES

170	T20 T20 BE
167	T20 T19 BE
164	T20 T18 BE
161	T20 T17 BE
158	T20 T20 D20
157	T20 T20 D19
156	T20 T20 D18
155	T20 T19 D19
154	T20 T18 D20
153	T20 T19 D18
152	T20 T20 D19
151	T20 T17 D20
149	T20 T19 D16
148	T20 T20 D16
147	T20 T17 D18
146	T20 T18 D16
145	T20 T19 D14
144	T20 T20 D12
143	T20 T17 D16
142	T20 T16 D20
141	T20 T19 D14
140	T20 T20 D16
139	T20 T13 D20
138	T20 T18 D12
137	T20 T19 D10
136	T20 T20 D8
135	BE T19 D14
134	T20 T16 D13
133	T20 T19 D8
132	T20 T16 D12
131	BE T19 BE
130	T20 T13 D16
129	T19 T16 D12
128	T19 T20 D6
127	T20 T17 D8

2 DART FINISHES

110	T20 BE
107	T19 BE
104	T18 BE
101	T17 BE
98	T20 D20
97	T20 D19
96	T20 D18
95	T19 D19
94	T18 D20
93	T19 D18
92	T20 D16
91	T17 D20
90	T18 D18
89	T19 D16
88	T20 D14
87	T17 D18
86	T18 D16
85	T19 D14
84	T18 D16
83	T19 D16
82	T20 D14
81	T17 D18
80	T18 D16
79	T19 D14
78	T18 D16
77	T17 D18



BELOW 99 THE GAME MAY BE FINISHED WITH ONLY 2 DARTS

Cases	Duplicated cases	1 st shot	2 nd shot	3 rd shot
1	1	180(T20x3)	180(T20x3)	141(2)
2	2,15	180(T20x3)	177(T20x2+T19x1)	144(1)
3	3,12,26	180(T20x3)	174(T20x2+T18x1)	147(2)
4	4,34	180(T20x3)	171(T20x2+T17x1)	150(2)
5	5,14,29,71	180(T20x3)	170(1)	151(2)
6	6,47	180(T20x3)	168(T20x2+T16x1)	153(1)
7	7,13,49,69	180(T20x3)	167(1)	154(2)
8	8,55	180(T20x3)	165(T20x2+T15x1)	156(1)
9	9,56,66	180(T20x3)	164(2)	157(1)
10	10,11,60,62	180(T20x3)	161(2)	160(1)
11	16	177(T20x2+T19x1)	177(T20x2+T19x1)	147(2)
12	17,27	177(T20x2+T19x1)	174(T20x2+T18x1)	150(2)
13	18,35	177(T20x2+T19x1)	171(T20x2+T17x1)	153(1)
14	19,25,40,70	177(T20x2+T19x1)	170(1)	154(2)
15	20,48	177(T20x2+T19x1)	168(T20x2+T16x1)	156(1)
16	21,24,50,67	177(T20x2+T19x1)	167(1)	157(1)
17	22,23,57,63	177(T20x2+T19x1)	164(2)	160(1)
18	28	174(T20x2+T18x1)	174(T20x2+T18x1)	153(1)
19	29,36	174(T20x2+T18x1)	171(T20x2+T17x1)	156(1)
20	30,41	174(T20x2+T18x1)	170(1)	157(1)
21	31,51,64	174(T20x2+T18x1)	167(1)	160(1)
22	32	174(T20x2+T18x1)	167(1)	160(1)
23	33,68	174(T20x2+T18x1)	170(1)	157(1)
24	37,38,42,65	171(T20x2+T17x1)	170(1)	160(1)
25	43,46,61	170(1)	170(1)	161(2)
26	44,45,52,54,58,59	170(1)	167(1)	164(2)
27	53	167(1)	167(1)	167(1)

1 st shot	2 nd shot	3 rd shot	Combine	Unique combinations
60x3	60x3	60x1+57x1+24x1 60x1+45x1+36x1	60x7+57x1+24x1 60x7+45x1+36x1	60x7+57x1+24x1 60x7+45x1+36x1
60x3	60x2+57x1	60x2+24x1	60x7+57x1+24x1	-
60x3	60x2+54x1	60x1+51x1+36x1 57x1+54x1+36x1	60x6+54x1+51x1+36x1 60x5+57x1+54x2+36x1	60x6+54x1+51x1+36x1 60x5+57x1+54x2+36x1
60x3	60x2+51x1	60x1+54x1+36x1 57x2+36x1	60x6+54x1+51x1+36x1 60x5+57x2+51x1+36x1	- 60x5+57x2+51x1+36x1
60x3	60x2+50x1	60x1+51x1+40x1 57x1+54x1+40x1	60x6+51x1+50x1+40x1 60x5+57x1+54x1+50x1+40x1	60x6+51x1+50x1+40x1 60x5+57x1+54x1+50x1+40x1
60x3	60x2+48x1	60x1+57x1+36x1	60x6+57x1+48x1+36x1	60x6+57x1+48x1+36x1
60x3	60x1+57x1+50x1	60x1+54x1+40x1 57x2+40x1	60x5+57x1+54x1+50x1+40x1 60x4+57x3+50x1+40x1	- 60x4+57x3+50x1+40x1
60x3	60x2+45x1	60x2+36x1	60x7+45x1+36x1	-
60x3	60x1+54x1+50x1 57x2+50x1	60x1+57x1+40x1	60x5+57x2+54x1+50x1+40x1 60x4+57x3+50x1+40x1	- -
60x3	60x1+51x1+50x1 57x1+54x1+50x1	60x2+40x1	60x6+51x1+50x1+40x1 60x5+57x1+54x1+50x1+40x1	- -
60x2+57x1	60x2+57x1	60x1+51x1+36x1 57x1+54x1+36x1	60x5+57x1+51x1+36x1 60x4+57x3+54x1+36x1	- 60x4+57x3+54x1+36x1
60x2+57x1	60x2+54x1	60x1+54x1+36x1 57x2+36x1	60x5+57x1+54x2+36x1 60x4+57x3+54x1+36x1	- -
60x2+57x1	60x2+51x1	60x1+57x1+36x1	60x5+57x2+51x1+36x1	-
60x2+57x1	60x2+50x1	60x1+54x1+40x1 57x2+40x1	60x5+57x1+54x1+50x1+40x1 60x4+57x3+50x1+40x1	- -
60x2+57x1	60x2+48x1	60x2+36x1	60x6+57x1+48x1+36x1	-
60x2+57x1	60x1+57x1+50x1	60x1+57x1+40x1	60x4+57x3+50x1+40x1	-
60x2+57x1	60x1+54x1+50x1 57x2+50x1	60x2+40x1	60x5+57x1+54x1+50x1+40x1 60x4+57x3+50x1+40x1	- -
60x2+54x1	60x2+54x1	60x1+57x1+36x1	60x5+57x1+54x2+36x1	-
60x2+54x1	60x2+51x1	60x2+36x1	60x6+54x1+51x1+36x1	-
60x2+54x1	60x2+50x1	60x1+57x1+40x1	60x5+57x1+54x1+50x1+40x1	-
60x2+54x1	60x1+57x1+50x1	60x2+40x1	60x5+57x1+54x1+50x1+40x1	-
60x2+54x1	60x1+57x1+50x1	60x2+40x1	60x5+57x1+54x1+50x1+40x1	-
60x2+54x1	60x2+50x1	60x1+57x1+40x1	60x5+57x1+54x1+50x1+40x1	-
60x2+51x1	60x2+50x1	60x2+40x1	60x6+51x1+50x1+40x1	-
60x2+50x1	60x2+50x1	60x1+51x1+50x1 57x1+54x1+50x1	60x5+51x1+50x3 60x4+57x1+54x1+50x3	60x5+51x1+50x3 60x4+57x1+54x1+50x3
60x2+50x1	60x1+57x1+50x1	60x1+54x1+50x1 57x2+50x1	60x4+57x1+54x1+50x3 60x3+57x3+50x3	- 60x3+57x3+50x3
60x1+57x1+50x1	60x1+57x1+50x1	60x1+57x1+50x1	60x3+57x3+50x3	-

(b) 13-combinations for 501 9-dart finish^[11]

그림 2. 501 9-다트 게임 종료 조합
Fig. 2. Combinations for 501 9-dart finish

있다.^[11] 여기서 3회 합계 점수의 체크아웃 표에서의 경우 수(빈도 수)를 고려하면 최종적으로 40가지를 얻는다. 표에서는 빈도수를 ‘()’안에 표기하였다. 이들 중 중복되는 것을 삭제하면 13개만 남는다.

이와 같이 최대 18가지 경우 수만 존재하는지 여부를 검증하기 위해 3장에서는 역 치환 알고리즘(RSA)을 제안하여 전수탐색(exhaustive search)하고자 한다.

III. 나눗셈 분기한정 알고리즘

다트게임의 획득 가능 점수(scores, s)는 $s_k = \{60, 57, 54, 51, 50, 48, 45, 42, 40, 39, 38, 36, 34, 33, 32, 30, 28, 27, 26, 25, 24, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1\}$ 로 $n=43$ 이다.

전형적인 $[0, 1]$ 배낭문제(knapsack problem, KP)에 대한 분기한정 법(BB)은 덧셈 또는 뺄셈을 하면서 깊이

우선탐색(depth-first search, DFS)로 이진트리의 레벨을 n 레벨까지 단계적으로 증가시키고 역추적(backtracking)하면서 전이진 트리를 만드는 방법으로 수행 복잡도는 $O(2^n)$ 으로 n 이 증가하면 지수적으로 증가하는 문제점을 갖고 있다. 다트 게임은 이와 같이 단순히 해당 물품을 추가 또는 제외의 $[0,1]$ 경우 수인 이진트리로 분기하는 방법이 아닌 하나의 부 노드에서 43개의 자 노드로 분기하므로 수행 복잡도는 $O(2^{43})$ 보다도 훨씬 복잡하여 컴퓨터 프로그램으로 해를 얻기도 쉽지 않다.

따라서 본 장에서는 $r_0 = 501$, $r_k - s_k$ 의 뺄셈이 아닌 $x_l = \lfloor \frac{r_k}{s_k} \rfloor$ 의 나눗셈으로 x_l 레벨로 점프하는 분기 한정법을 제안하며, 레벨 l 은 8까지로 한정시키고, 레벨 7에서 레벨 8의 자 노드로 분기하는 가지 수는 43개가 아닌 $r_j = s_L + s_U$ 의 $[s_L, s_U]$ 범위로 한정하여 빠르게 해를 얻는 방법을 채택한다. 이를 나눗셈 분기 한정 알고리즘(DBBA)이라 하며 다음과 같이 수행된다.

본 장에서는 DBBA를 트리를 이용하는 방법이 아닌 행렬을 이용하는 방법을 적용하여, $x_j \geq 2$ 를 갖는 점수들에 대해 마지막 점수부터 x_j 를 '1'씩 점진적으로 감소시키면서 나머지에 대한 가능한 점수 조합을 찾는다.

획득 점수 $s_k = \{60, 57, 54, 51, 50, 48, 45, 42, 40, 39, 38, 36, 34, 33, 32, 30, 28, 27, 26, 25, 24, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1\}$ 로 설정한다.

501점을 획득하여야만 경기가 종료되는 주어진 문제에서 최소의 다트 축을 던지는 횟수 z 인 퍼펙트게임은 $S = 501$, $\max s_j = 60$ 에 대해 $z = \lfloor \frac{S}{\max s_j} \rfloor = 501/60 = \lceil 8.35 \rceil = 9$ 회이다. 따라서 조합 행렬의 j 열을 다음과 같이 배정한다.

Step 1. $z = 9$ 로 j 열에는 1, 2, ..., 9를 배정한다.

i 행에는 $s_k \geq S/z$ 를 배정한다.

Step 2. i 행에는 $\frac{501}{9} = 55.67$ 로 $s_i \geq 56$ 으로 $56 \leq s_i \leq 60$ 인 $s_i = \{60, 57\}$ 를 배정한다.

Step 3. $l = \lfloor \frac{S}{s_i} \rfloor$ 의 l 개인 s_i 를 x_{ij} 에 배정한다.

$S - s_i \times l$ 인 나머지가 s_k 에 존재하면 이를 $j = l + 1$ 에 x_{ij} 로 배정한다. [트리 좌측 가지 레벨 8과 우측 가지 레벨 8로 점프]

이와 같이 수행하면 $501 = 60 \times 8 + 21$, $501 = 57 \times 8 + 45$ 로 $s_i = 60$ 과 57에 대해 다음과 같이 행렬에 x_i 이 채워진다.

	1	2	3	4	5	6	7	8	9
60	60	60	60	60	60	60	60	60	21
57	57	57	57	57	57	57	57	57	45

Step 4. $s_i = \{60, 57\}$ 각각에 대해 $x_j = 8$ 이므로 $j = 8, 7, \dots, 2$ 의 $s_j = s_i$ 을 삭제한 경우 $\min(s_j - s_2)$ 인 부족 점수 r_i 를 채울 수 있는 빈 셀의 가능한 점수 조합 범위 $[s_1, s_2]$ 를 결정하고, s_2 내림차순으로 가능한 점수들을 채워 넣는다. $s_1 \geq s_2$ 가 되면 이전의 i 조합에서 $j = j - 1$ 에 대해 이 과정을 반복한다. [역 추적으로 이전 레벨의 해 생성]

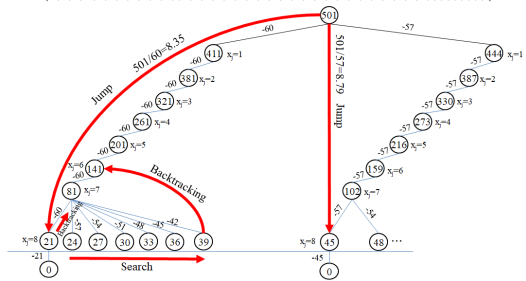
추가되는 각 경우 수의 비어있는 셀을 채우는 방법은 나머지(modulo) 또는 부족점수인 $r_j = S - s_i x_j$ 를 비어있는 셀의 수를 충족하는 점수 조합을 s_k 에서 찾아 추가한다. 만약 나머지 r_j 를 충족하는 비어 있는 j 열 셀의 개수만큼 $r_j < s_i$ 이면 단일 점수를 찾고, $r_j > s_i$ 이면 j 열의 2개 점수 합인 $r_j = s_1 + s_2$, $s_k < s_i$ 인 $s_k x_k$ 또는 $s_k x_k + s_k$ 가 되며, 만약 두 점수 $s_1 < s_2$ 가 선택된 경우 $s_1 < s_a < s_b < s_2$ 의 s_a, s_b 쌍에 대해 $s_a = s_b$ 가 될 때까지 경우 수를 계속적으로 추가한다. 여기서 s_2 는 $\min(s_j - s_2)$ 이 되는 수에 대해 $r_j = s_2 - s_1$ 조건을 충족하여야만 한다.

제안된 DBBA를 $s_i = 60$ 의 $x_j = 8$ 로부터 $x_j = 7$ 인 경우 가능한 조합 경우 수를 찾는 과정은 그림 3과 같다.

$s_i = 60$	1	2	3	4	5	6	7	8	9	나머지
$x_j = 8$	60	60	60	60	60	60	60	60	21	21
$x_j = 7$	60	60	60	60	60	60	60	57	24	81
	60	60	60	69	60	60	60	54	27	
	60	60	60	60	60	60	60	51	30	
	60	60	60	60	60	60	60	48	33	
	60	60	60	60	60	60	60	45	36	
	60	60	60	60	60	60	60	42	39	

(a) Matrix

Scores = {60, 57, 54, 51, 50, 48, 45, 42, 40, 39, 38, 34, 33, 32, 30, 28, 27, 26, 25, 24, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1}



(b) Tree

그림 3. $s_i = 60$, $x_j = 7$ 에 대한 DBBA

Fig. 3. DBBA for $s_i = 60$, $x_j = 7$

표 1. $C=501$, $z=9$ 의 퍼펙트 닳트게임 점수 경우 수

Table 1. Perfect dart game score cases for $C=501$, $z=9$

Cases	#1	#2	#3	#4	#5	#6	#7	#8	#9	Scores	$s_j \cdot x_j$	Frequency	Multiply	
1	60	60	60	60	60	60	60	60	21	501	60x8+21x1	1 1	1	
2	60	60	60	60	60	60	60	57	24	501	60x7+57x1+24x1	1 1 2	2	
3	60	60	60	60	60	60	60	54	27	501	60x7+54x1+27x1	1 1 1	1	
4	60	60	60	60	60	60	60	51	30	501	60x7+51x1+30x1	1 1 2	2	
5	60	60	60	60	60	60	60	48	33	501	60x7+48x1+33x1	1 1 1	1	
6	60	60	60	60	60	60	60	45	36	501	60x7+45x1+36x1	1 1 2	2	
7	60	60	60	60	60	60	60	42	39	501	60x7+42x1+39x1	1 1 1	1	
8	60	60	60	60	60	60	57	57	27	501	60x6+57x2+27x1	1 1 1	1	
9	60	60	60	60	60	60	57	54	30	501	60x6+57x1+54x1+30x1	1 1 1 2	2	
10	60	60	60	60	60	60	57	51	33	501	60x6+57x1+51x1+30x1	1 1 1 1	1	
11	60	60	60	60	60	60	57	50	34	501	60x6+57x1+50x1+34x1	1 1 1 1	1	
12	60	60	60	60	60	60	57	48	36	501	60x6+57x1+48x1+36x1	1 1 1 2	2	
13	60	60	60	60	60	60	57	45	39	501	60x6+57x1+45x1+39x1	1 1 1 1	1	
14	60	60	60	60	60	60	57	42	42	501	60x6+57x1+42x2	1 1 1	1	
15	60	60	60	60	60	60	54	54	33	501	60x6+54x2+33x1	1 1 1	1	
16	60	60	60	60	60	60	54	51	36	501	60x6+54x1+51x1+36x1	1 1 1 2	2	
17	60	60	60	60	60	60	54	48	39	501	60x6+54x1+48x1+39x1	1 1 1 1	1	
18	60	60	60	60	60	60	54	45	42	501	60x6+54x1+45x1+42x1	1 1 1 1	1	
19	60	60	60	60	60	60	51	51	39	501	60x6+51x2+39x1	1 1 1	1	
20	60	60	60	60	60	60	51	50	40	501	60x6+51x1+50x1+40x1	1 1 1 1	1	
21	60	60	60	60	60	60	51	48	42	501	60x6+51x1+48x1+42x1	1 1 1 1	1	
22	60	60	60	60	60	60	51	45	45	501	60x6+51x1+45x2	1 1 1	1	
23	60	60	60	60	60	60	48	48	45	501	60x6+48x2+45x1	1 1 1	1	
24	60	60	60	60	60	57	57	57	30	501	60x5+57x3+30x1	1 1 2	2	
25	60	60	60	60	60	57	57	54	33	501	60x5+57x2+54x1+33x1	1 1 1 1	1	
26	60	60	60	60	60	57	57	51	36	501	60x5+57x2+51x1+36x1	1 1 1 2	2	
27	60	60	60	60	60	57	57	48	39	501	60x5+57x2+48x1+39x1	1 1 1 1	1	
28	60	60	60	60	60	57	57	45	42	501	60x5+57x2+45x1+42x1	1 1 1 1	1	
29	60	60	60	60	60	57	54	54	36	501	60x5+57x1+54x2+36x1	1 1 1 2	2	
30	60	60	60	60	60	57	54	51	39	501	60x5+57x1+54x1+51x1+39	1 1 1 1 1	1	
31	60	60	60	60	60	57	54	50	40	501	60x5+57x1_54x1_50x1+40x1	1 1 1 1 1	1	
32	60	60	60	60	60	57	54	48	42	501	60x5+57x1+54x1+48x1+42x1	1 1 1 1 1	1	
33	60	60	60	60	60	57	54	45	45	501	60x5+57x1+54x1+45x2	1 1 1 1	1	
34	60	60	60	60	60	57	51	48	45	501	60x5+57x1+51x1+48x1+45x1	1 1 1 1 1	1	
35	60	60	60	60	60	54	51	51	45	501	60x5+54x1+51x2+45x1	1 1 1 1	1	
36	60	60	60	60	60	54	51	48	48	501	60x5+54x1+51x1+48x2	1 1 1 1	1	
37	60	60	60	60	60	51	50	50	50	501	60x5+51x1+50x3	1 1 1	1	
38	60	60	60	60	57	57	57	57	33	501	60x4+57x4+33x1	1 1 1	1	
39	60	60	60	60	57	57	57	54	36	501	60x4+57x3+54x1+36x1	1 1 1 2	2	
40	60	60	60	60	57	57	57	51	39	501	60x4+57x3+51x1+39x1	1 1 1 1	1	
41	60	60	60	60	57	57	57	50	40	501	60x4+57x3+50x1+40x1	1 1 1 1	1	
42	60	60	60	60	57	57	57	48	42	501	60x4+57x3+48x1+42x1	1 1 1 1	1	
43	60	60	60	60	57	57	57	45	45	501	60x4+57x3+45x2	1 1 1	1	
44	60	60	60	60	57	57	54	54	39	501	60x4+57x2+54x2+39x1	1 1 1 1	1	
45	60	60	60	60	57	57	54	51	42	501	60x4+57x2+54x1+51x1+42x1	1 1 1 1	1	
46	60	60	60	60	57	57	54	48	45	501	60x4+57x2+54x1+48x1+45x1	1 1 1 1	1	
47	60	60	60	60	57	57	51	51	45	501	60x4+57x2+51x2+45x1	1 1 1 1	1	
48	60	60	60	60	57	57	51	48	48	501	60x4+57x2+51x1+48x2	1 1 1 1	1	
49	60	60	60	60	57	54	54	54	42	501	60x4+57x1+54x3+42x1	1 1 1 1	1	
50	60	60	60	60	57	54	54	51	45	501	60x4+57x1+54x2+51x1+45x1	1 1 1 1 1	1	
51	60	60	60	60	57	54	54	48	48	501	60x4+57x1+54x2+48x2	1 1 1 1	1	
52	60	60	60	60	57	54	51	51	48	501	60x4+57x1+54x1+51x2+48x1	1 1 1 1 1	1	
53	60	60	60	60	57	54	50	50	50	501	60x4+57x1+54x1+50x3	1 1 1	1	
54	60	60	60	60	54	54	54	51	48	501	60x4+54x3+51x1+48x1	1 1 1 1 1	1	
55	60	60	60	60	54	54	51	51	51	501	60x4+54x2+51x3	1 1 1 1	1	
56	60	60	60	60	57	57	57	57	36	501	60x3+57x5+36x1	1 1 2	2	
57	60	60	60	60	57	57	57	54	39	501	60x3+57x4+54x1+39x1	1 1 1 1	1	
58	60	60	60	60	57	57	57	51	42	501	60x3+57x4+51x1+42x1	1 1 1 1	1	
59	60	60	60	60	57	57	57	48	45	501	60x3+57x4+48x1+45x1	1 1 1 1	1	
60	60	60	60	60	57	57	57	54	54	42	501	60x3+57x3+54x2+42x1	1 1 1 1	1
61	60	60	60	60	57	57	57	54	48	48	501	60x3+57x3+54x1+48x2	1 1 1 1	1
62	60	60	60	60	57	57	57	51	51	48	501	60x3+57x3+51x2+48x1	1 1 1 1	1

Cases	#1	#2	#3	#4	#5	#6	#7	#8	#9	Scores	$s_j \cdot x_j$	Frequency	Multiply
63	60	60	60	57	57	57	50	50	50	501	$60x3+57x3+50x3$	1 1 1	1
64	60	60	60	57	57	54	54	54	45	501	$60x3+57x2+54x3+45x1$	1 1 1 1	1
65	60	60	60	57	57	54	54	51	48	501	$60x3+57x2+54x2+51x1+48x1$	1 1 1 1 1	1
66	60	60	60	57	57	54	51	51	51	501	$60x3+57x2+54x1+51x3$	1 1 1 1	1
67	60	60	60	57	54	54	54	54	48	501	$60x3+57x1+54x4+48x1$	1 1 1 1	1
68	60	60	60	57	54	54	54	51	51	501	$60x3+57x1+54x3+51x2$	1 1 1 1	1
69	60	60	60	54	54	54	54	54	51	501	$60x3+54x5+51x1$	1 1 1	1
70	60	60	57	57	57	57	57	57	39	501	$60x2+57x5+39x1$	1 1 1	1
71	60	60	57	57	57	57	57	54	42	501	$60x2+57x5+54x1+42x1$	1 1 1 1	1
72	60	60	57	57	57	57	57	51	45	501	$60x2+57x5+51x1+45x1$	1 1 1 1	1
73	60	60	57	57	57	57	57	48	48	501	$60x2+57x5+48x2$	1 1 1	1
74	60	60	57	57	57	57	54	54	45	501	$60x2+57x4+54x2+45x1$	1 1 1 1	1
75	60	60	57	57	57	57	54	51	48	501	$60x2+57x4+54x1+51x1+48x1$	1 1 1 1	1
76	60	60	57	57	57	57	51	51	51	501	$60x2+57x4+51x3$	1 1 1	1
77	60	60	57	57	57	54	54	54	48	501	$60x2+57x3+54x3+48x1$	1 1 1 1	1
78	60	60	57	57	57	54	54	51	51	501	$60x1+57x3+54x2+51x2$	1 1 1 1	1
79	60	60	57	57	54	54	54	54	51	501	$60x2+57x2+54x4+51x1$	1 1 1 1	1
80	60	57	57	57	57	57	57	57	42	501	$60x1+57x7+42x1$	1 1 1	1
81	60	57	57	57	57	57	57	54	45	501	$60x1+57x6+54x1+45x1$	1 1 1 1	1
82	60	57	57	57	57	57	57	51	48	501	$60x1+57x6+51x1+48x1$	1 1 1 1	1
83	60	57	57	57	57	57	54	54	48	501	$60x1+57x5+54x2+48x1$	1 1 1 1	1
84	60	57	57	57	57	57	54	51	51	501	$60x1+57x5+54x1+51x2$	1 1 1 1	1
85	60	57	57	57	57	54	54	54	51	501	$60x1+57x4+54x3+51x1$	1 1 1 1	1
86	60	57	57	57	54	54	54	54	54	501	$60x1+57x3+54x5$	1 1 1	1
87	57	57	57	57	57	57	57	57	45	501	$57x8+45x1$	1 1	1
88	57	57	57	57	57	57	57	51	51	501	$57x7+51x2$	1 1	1
89	57	57	57	57	57	57	54	54	51	501	$57x6+54x2+51x1$	1 1 1	1
90	57	57	57	57	57	54	54	54	54	501	$57x5+54x4$	1 1	1
Total													101

$60x8+21x1=501$ 의 $s_i = 60$, $x_j = 8$ 의 최대 x_j 에서 $x_j = 8 - 1 = 7$ 로 되면 $501 - 60x7 = 81$ 의 나머지(부족점수)가 된다. 이를 채울 2개 점수의 조합은 $81/2 = 40.5$ 의 점수가 존재하지 않아 $\min(s_j - s_2)$ 인 $s_2 = 57$ 로 결정되었으며, $81 - 57 = 24$ 로 s_k 에 24(T8)이 존재하여 $x_j = 7$ 에서 가능한 조합 경우 수는 [24, 57]이다. 따라서 [24, 57] 범위 점수들인 $s_k = \{57, 54, 51, 50, 48, 45, 42, 40, 39, 38, 36, 34, 33, 32, 30, 28, 27, 26, 25, 24\}$ 의 점수들에 대해 합이 '81'이 되는 쌍을 s_1 을 증가, s_2 를 감소하면서 $s_1 \geq s_2$ 가 될 때까지 찾은 결과가 그림 3에 제시되어 있다. 이와 같은 방법으로 $s_i = 60$, $x_j = 8$ 에 대해 $x_j = 7, 6, 5, 4, 3, 2, 1$ 에 대한 가능한 조합 경우 수를 찾으면 전수탐색을 할 수 있다. $s_k = \{57, 54, 51, 50, 48, 45, 42, 40, 39, 38, 36, 34, 33, 32, 30, 28, 27, 26, 25, 24\}$ 에 대해 두 점수 쌍 $(s_2, s_1), s_2 \geq s_1$ 의 합이 81이 되는 가능한 조합의 경우 수를 찾아보면 $(57, 24)=81$, $(54, 27)=81$, $(51, 30)=81$, $\cancel{/(50, 31)=81}$, $(48, 33)=81$, $(45, 36)=81$, $(42, 39)=81$, $\cancel{/(40, 41)=81}$ ($s_2 < s_1$)이다. 여기서 $\cancel{/(50, 31)=81}$ 의 경우 $s_2 = 50$ 에 대한 $s_1 = 31$ 은 존재하지 않아 제외되고 $\cancel{/(40, 41)=81}$ 역시 $s_2 < s_1$ 인 동시에 $s_1 = 41$ 이 존재하지

않아 제외된다. 결국 $s_i = 60$, $x_j = 7$ 인 $60x7 + s_2x1 + s_1x1$ 의 해는 $(57, 24)=81$, $(54, 27)=81$, $(51, 30)=81$, $(48, 33)=81$, $(45, 36)=81$, $(42, 39)=81$ 의 6가지가 존재함을 알 수 있다.

IV. 적용 및 결과 분석

본 장에서는 3장에서 제안된 나눗셈 분기 한정 알고리즘(DBBA)을 적용하여 $C = 501$ 의 $z = 9$ 에 대한 가능한 경우 수를 표 1과 같이 찾을 수 있었다.

기준에 알려진 점수 기준 경우 수는 18개 또는 40개이었으나 본 역 치환 알고리즘으로 찾은 점수 기준 서로 다른 경우 수는 90개이며, 다른 과녁에 맞추어도 동일 점수를 획득할 수 있는 관계로 각 점수의 빈도수를 고려하면 총 조합의 경우 수는 101개임을 알 수 있다. 이 결과는 다음과 같이 계산되었다.

일예로 $60x8+21x1$ 의 경우 $60(T20)$, $21(T7)$ 로 $T20x8 + T7x1$ 로 계산되어 점수 경우 수=1, 조합 경우 수(빈도수 곱, frequency multiply)=1로 계산된다.

$60x7+57x1+24x1$ 의 경우 $60(T20)$, $57(T19)$,

24(D12, T8)로 T20x7+T19x1+D12x1과 T20x7+T19x1+T8x1로 획득 가능하여 점수 기준 경우 수는 1이지만 조합 경우 수는 2로 계산된다.

V. 결 론

본 논문은 $S=501$ 점의 점수를 획득하여 퍼펙트 다트 게임의 승자가 되기 위해 9회의 다트 축을 던지는 경우 가능한 획득 점수의 조합 경우 수가 몇 개인지 정확히 찾아내는 알고리즘을 제안하였다.

제안된 알고리즘은 다트게임에서 획득할 수 있는 최고 점수인 $\max s_k=60$ 점(T20)을 대상으로 $z = \lceil \frac{S}{\max s_j} \rceil = 501/60 = \lceil 8.35 \rceil = 9$ 를 얻었으며, $\frac{501}{9} = 55.67$ 로 $s_i \geq 56$ 으로 $56 \leq s_i \leq 60$ 인 $s_i = \{60, 57\}$ 에 대해 $501 = 60 \times 8 + 21 \times 1$, $501 = 57 \times 8 + 45 \times 1$ 의 기본 조합을 계산하였다.

이 기본 조합에 대해 $x_j = 8$ 에 대해 7, 6, ..., 2로 감소시키면서 9개 셀의 비어 있는 셀들에 부족 점수를 채울 수 있는 점수 쌍 또는 조합을 구하였다. 이 점수 쌍 범위 $[s_1, s_2], s_1 < s_2$ 에 대해 $s_2 = s_1$ 이 될 때까지 가능한 조합을 찾는 나눗셈 분기한정 알고리즘을 제안하였다.

제안된 알고리즘을 501 9-다트 퍼펙트게임에 적용한 결과 기존에 알려진 18가지 경우 수에 비해 101가지 경우 수가 있음을 보였다.

특이한 점은 기존에는 T20(60)점에 대한 가능한 조합만을 제시한데 반해 제안된 알고리즘은 T20(60)에 대한 97개와 T19(57)에 대한 4개도 함께 찾아낼 수 있었다.

References

[1] Wikipedia, "Nine-Dart Finish," https://en.wikipedia.org/wiki/Nine-dart_finish, Retrieved Feb. 2022.

[2] Wikipedia, "Darts," <https://en.wikipedia.org/wiki/Darts#Scoring>, Retrieved Feb. 2022.

[3] Lardyarts, "Getting a Nine Dart Finish: How to Throw Like a Pro and Get a Nine Dart Finish," <https://www.lardyarts.com/how-to/9-dart-finish>, Retrieved Feb. 2022.

[4] D. Boyle, "In Search of the Nine Dart Finish: Understanding Standards in a Time of Frustration," <https://www.linkedin.com/pulse/search-nine-dart-finish-unrelenting-standards-time-david-boyle>, Nov. 2020.

[5] P. Lees, "Possible Mathematical Finishes to the Darts Game 501," <https://www.titanwolf.org/Network/q/fbfbddde-6f93-4348-b443-ac3c664e598c/x>, Retrieved Feb. 2022.

[6] Jesscowie96, "The Math Behind the Perfect Nine-Dart Finish," <https://shipmitreb.wordpress.com/2017/03/08/the-math-behind-the-perfect-nine-dart-finish/>, Mar. 2017.

[7] Northstandchat, "Thread: 9 Dart Finish Combinations," <https://www.northstandchat.com/showthread.php?140891-9-dart-finish-combinations>, Retrieved Feb. 2022.

[8] Quora, "How do You Get a 9 Dart Finish?," <https://www.quora.com/How-do-you-get-a-9-dart-finish>, Retrieved Feb. 2022.

[9] Idswater, "How Many Combinations does a 9 Darter Have?," <https://idswater.com/2020/02/22/how-many-combinations-does-a-9-darter-have/>, Feb. 2020.

[10] Dartworld, "Dart World Q and A: Helpful Information," <https://www.dartworld.com/dart-world-questions-answers.html>, Retrieved Feb. 2022.

[11] Dartn, "Alle 71 verschiedenen Wege zum 9-Darter," https://www.dartn.de/Liste_aller_9-Darter_Moeglichkeiten, Retrieved Feb. 2022.

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 ~ 2007.2 : 국립 원주대학 여성교양과 조교수
- 2007.3 ~ 2015.3 : 강릉원주대학교 멀티미디어공학과 부교수
- 2015.4 ~ 현재 : 강릉원주대학교 멀티미디어공학과 정교수
- 관심분야 : 소프트웨어 프로젝트 관리, 개발 방법론, 분석과 설계 방법론, 시험 및 품질보증, 소프트웨어 신뢰성, 인공지능과 빅데이터분석, 최적화 알고리즘
- e-mail : sulee@gwnu.ac.kr