Naval Research Laboratory
Washington, DC 20375-5320

NRL/5340/MR—2021/6

The Number of Joint Association Events in the JPDAF Accounting for Gating

DAVID FREDERIC CROUSE

Surveillance Technology Branch
Radar Division

September 7, 2021

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
07-09-2021

2. REPORT TYPE
NRL Memorandum Report

3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE

The Number of Joint Association Events in the JPDAF Accounting for Gating

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

David Frederic Crouse

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
1X45

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/5340/MR--2021/6

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
One Liberty Center
875 N. Randolph Street, Suite 1425
Arlington, VA 22203-1995

10. SPONSOR / MONITOR'S ACRONYM(S)

ONR

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Variants of the Joint Probabilistic Data Association Algorithm Filter (JPDAF) for target tracking have been around for years. The most computationally-demanding step of the algorithm is the computation of target-measurement association probabilities, which can involve summing over joint association hypotheses of targets to measurements and missed detections. This report goes beyond the worst-case approximations in the literature and provides a simple explicit expression for the total number of joint association events given a 0-1 matrix specifying which targets gate with which measurements.

15. SUBJECT TERMS

Data association   Complexity   JPDA
JPDAF   Matrix permanent

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
UU	UU	UU	UU	9	David F. Crouse
					19b. TELEPHONE NUMBER *(include area code)* (202) 404-1859

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

This page intentionally left blank.

THE NUMBER OF JOINT ASSOCIATION EVENTS IN THE JPDAF ACCOUNTING FOR GATING

1. INTRODUCTION

The Joint Probabilistic Data Association Filter (JPDAF), the multi-target version of the probabilistic data association filter (PDAF), has been used for target tracking in numerous radar systems around the world, some of which are named in [1]. One of the first and best-known examples is Australia's Jindalee radar for over-the-horizon tracking [1, 2]. The idea behind the filter is fairly simple: Given a set of predicted target state estimates, and a set of measurements, one wishes to determine a posterior set of target state estimates. However, one does not know which measurement goes with which target. Thus, one approximates the posterior PDF of each target state as a Gaussian distribution whose mean is the expected value of the posterior distribution and whose covariance matrix is the covariance of the posterior distribution. Track initiation and termination are handled separately, such as using cascaded logic [3, Ch. 3.3] or in an "integrated" variant of the algorithm [4–6].

The computation of the mean requires the computation of a set of target-measurement association probabilities. In practice, variants of the algorithm seldom use the mean value and it has been demonstrated that the use of the mean leads to an undesirable track coalescence [7, 8]. However, alternative algorithms often have to compute target-measurement association probabilities, as is the case in the global-nearest-neighbor (GNN) JPDAF described in [9].

The computation of the target-measurement association probabilities can be performed through enumeration of all possible hypotheses of measurements assigned to targets and missed detections. However, computational efficiency can be improved by taking advantage of the structure of the problem. At the simplest level, gating can eliminate the possibility of assigning a measurement to a particular target. However, greater computational efficiencies can be achieved by taking advantage of the structure of the problem regardless of gating, such as through the use of a matrix permanent formulation in [10], with a factored hypothesis enumeration techniques that can make use of gating information, as in [11], or by combining the gates of targets into a network as in [12, 13],[1] among other methods. Additionally, the computational complexity can be further reduced by clustering targets that gate with common measurements, as in [16]. However, the total number of possible joint association events remains a reasonable approximation to the overall difficulty of an association problem.

In [17], the total number of possible joint-measurement association events is derived assuming that all targets gate with all measurements. The same solution was derived again in [18]. The result is summarized in Section 2.

In [17], the relation between determining the number of joint association events and the problem of generating constrained combinatorial arrangements is noted. Methods for estimating the total number of

Manuscript approved August 30, 2021.

[1]The algorithm of [12] is patented in [14] (The patent should be expiring in a few years. Note that the patent cited in the paper [12] itself was not granted [15].

constrained arrangements are given in [19, Ch. 9] and are very complicated. Section 3 of this report provides a simple expression for the total number of joint association events that does not involve generating every single event. Rather the matrix permanent is used. Section 4 then provides two examples and concludes the paper.

2. THE NUMBER OF HYPOTHESES WITHOUT GATING

Let N_T be the number of targets and N_M be the number of measurements. As given in [17, 18], under the constraint that a valid joint association event is such that

1. Each of the N_T targets is assigned either to a measurement or declared not detected.
2. Each of the N_M measurements is assigned to at most one target.
3. No two measurements can be assigned to the same target.
4. No two targets are assigned to the same measurement.

the total number of joint association events, is

$$\begin{aligned}
N_{Hyps} &= \underbrace{\sum_{n=0}^{\min(N_T,N_M)}}_{\substack{\text{Sum over the number}\\ \text{of target observed}}} \underbrace{\binom{N_T}{n}}_{\substack{\text{Choose which targets}\\ \text{are observed}}} \underbrace{\binom{N_M}{n}}_{\substack{\text{Choose which measurements}\\ \text{are from targets}}} \underbrace{n!}_{\substack{\text{Assign measurements}\\ \text{to targets}}} && (1)\\
&= \sum_{n=0}^{\min(N_T,N_M)} \frac{N_T!N_M!}{n!(N_T-n)!(N_M-n)!} && (2)\\
&= \sum_{n=0}^{\min(N_T,N_M)} c_n && (3)
\end{aligned}$$

where the $c_0 = 1$ and the c_n term can be recursively computed as

$$c_n = c_{n-1}\frac{(N_M-n+1)(N_T-n+1)}{n} \tag{4}$$

Equation (3) is implemented as the `num2DTarMeasHyps` function in the Tracker Component Library [20, 21].

3. THE NUMBER OF HYPOTHESES WITH GATING

Gating is a technique used to eliminate the possibility that a particular measurement originated from a particular target, thus reducing the complexity of computing the target-measurement association probability. If measurements are too far away from a target according to a particular measure, then they are not considered. Usually, the Mahalanobis distance is used, as in [3, Ch. 2.3].

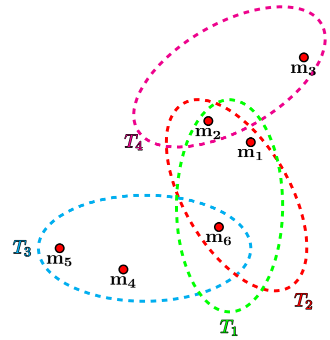

Fig. 1—An example showing potential measurements (**m** values) falling within gating ellipses of targets (colored T values). The corresponding gating matrix between targets and measurements is given in Equation (5).

As an example, consider the illustration of (1), where there are four targets and six measurements. Here, the gating regions of the targets are represented as fixed ellipses (whereas the use of Mahalanobis distances would have created differing-sized regions based on the covariance matrices of the measurements). Letting the rows of a binary matrix represent target and the columns measurements, the corresponding gating matrix is:

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{5}$$

where a 1 indicates that a measurement gates with a target and a 0 indicates that it does not. For example, no valid joint association events contain an assignment between target 4 and measurement, because they do not gate together.

To solve the problem for a general gating matrix $\mathbf{X}$, we augment the matrix as

$$\mathbf{X}_{\text{aug}} = \begin{bmatrix} \mathbf{X} & \mathbf{I} \end{bmatrix} \tag{6}$$

where $\mathbf{I}$ is a square identity matrix. This type of matrix augmentation is analogous to what was done when augmenting the cost matrix in the rectangular 2D assignment algorithm implementation described in [22]: it introduces the missed detection assignments into the problem.

Possible joint association events are those such that

1. Each of the N_T rows is assigned to a column with a 1 in it.
2. Each of the N_M columns is assigned to at most one row.
3. No two columns can be assigned to the same row.

4. No two rows are assigned to the same column.

The assignment condition in 1 is such that the target rows are either assigned to a measurement column, or to one of the columns of the appended identity matrix, which functions as a set of missed detection hypotheses.

A brute-force way to count the number of joint association events is thus

$$N_{Hyps} = \sum_{\sigma \in P(N_T+N_M, N_T)} \prod_{i=1}^{N_T} x_{i,\sigma_i}^{\text{aug}} \tag{7}$$

where $x_{i,j}^{\text{aug}}$ is the entry in row i and column j of $\mathbf{X}_{\text{aug}}$, $P(a,b)$ is the set of permutations of length a from the set $\{1,\ldots,b\}$ (also known as arrangements of b into a parts), and σ_i is the ith entry of the arrangement vector $\boldsymbol{\sigma}$. if any of the x^{aug} terms in the product in (7) are zero, then the joint association event is invalid and is not counted. However, this is just the definition of the matrix permanent:

$$N_{Hyps} = \text{perm}\left(\mathbf{X}_{\text{aug}}\right) \tag{8}$$

The matrix permanent can be exactly computed in significantly less time than the brute-force expression of (7) using various algorithms, including [23, Ch. 2, Theorem 4.1, pg. 26], [24] and [25, Ch. 23]. An implementation is given in the `perm` function in the Tracker Component Library [20, 21].

However, those algorithms scale non-polynomialy with the number of elements in the array. It has been shown that the computation of the matrix permanent of a general matrix filled with 0's and 1's is #P-Complete [26, 27], which means that it is probably not possible to find an algorithm that can produce an exact solution in polynomial time with respect to the size of the input matrix, though low-complexity approximations exist [28]. Consequently, if it is possible to find a polynomial time algorithm to solve (8), either one must prove that #P-Complete problems can be completed in polynomial time, or the solution would have to make use of the fact that $\mathbf{X}_{\text{aug}}$ is not a general 0-1 matrix, but rather it has an identity matrix appended to the end.

4. EXAMPLES AND CONCLUSION

The number of hypotheses with and without gating can differ by a large amount. For example, with the $N_T = 4$ and $N_M = 6$ example gating matrix in (5), the worst-case number of solution without gating from (3) is 1045. However, the total number of gates hypotheses is only 116. Also, though the complexity scales exponentially, Ryser's algorithm [23, Ch. 2, Theorem 4.1, pg. 26], which is implemented in the Tracker Component Library can still solve some problems much faster than brute force.

For example, consider the following 12×20 gating matrix:

$$
\mathbf{X} = \begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1
\end{bmatrix} \tag{9}
$$

The upper bound on the number of solutions is 201301915072081. The solution as obtained using the matrix permanent is 200329623077. When run in Matlab using the compiled version of the `perm` function in the Tracker Component Library on a computer with a 2.6 GHz processor, the solution took about 25 seconds to obtain. On the other hand, if one were to visit every one of the valid combinations per clock cycle of the processor, it would have taken over 21 hours.

Additionally, the speed of the matrix permanent algorithm of [23, Ch. 2, Theorem 4.1, pg. 26] does not depend on the values in the matrix, only on the size of the matrix. Thus, if $\mathbf{X}$ were all ones, it would take about the same amount of time, whereas a brute-force visit of all 201301915072081 joint association events with one per clock cycle of a 2.6 GHz processor would take a bit less than two and a half years to complete.

That said, the matrix permanent is not fast. It is thus not practicable to utilize an exact matrix permanent to analyze the complexity of a JPDAF assignment problem to determine whether or not one wishes to solve it exactly or with an approximation. However, for such an application, bounds on the matrix permanent, such as those in [28], might be useful. Alternatively, one could just set maximum limits based on the dimensions of the gating matrix.

REFERENCES

1. Y. Bar-Shalom and F. Daum, "The Probabilistic Data Association Filter," *IEEE Control Systems Magazine* pp. 82–100 (Dec. 2009).

2. S. B. Colegrove, "Advanced Jindalee Tracker: Probabilistic Data Association Multiple Model Initiation Filter," DSTO-TR-0659, Department of Defence, Defense Science and Technology Organization, Salisbury, SA, Australia, 1999.

3. Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms* (YBS Publishing, Storrs, CT, 2011).

4. D. Musicki and R. Evans, “Joint Integrated Probabilistic Data Association: JIPDA,” *IEEE Transactions on Aerospace and Electronic Systems* **40**(3), 1093–1099 (July 2004).

5. S. Challa, M. R. Morelande, D. Musicki, and J. Evans, *Fundamentals of Object Tracking* (Cambridge Unviersity Press, Cambridge, 2011).

6. J. L. Williams, “Marginal multi-Bernoulli Filters: RFS Derivation of MHT, JIPDA, and association-based MeMBer,” *IEEE Transactions on Aerospace and Electronic Systems* **51**(3), 1664–1687 (July 2015).

7. D. F. Crouse, “Advances in Displaying Uncertain Estimates of Multiple Targets,” Proceedings of the Proceedings of SPIE: Signal Processing, Sensor Fusion, and Target Recognition XXII, volume 8745, Baltimore, MD, 23 May 2013.

8. R. J. Fitzgerald, “Track Biases and Coalescence with Probabilistic Data Association,” *IEEE Transactions on Aerospace and Electronic Systems* **21**(6), 822–825 (Nov. 1985).

9. O. E. Drummond, “Best Hypothesis Target Tracking and Sensor Fusion,” Proceedings of the Proceedings of SPIE: Signal and Data Processing of Small Targets, volume 3809, Denver, CO, July 1999, pp. 586–600.

10. D. F. Crouse and P. Willett, “Computation of Target-Measurement Association Probabilities Using the Matrix Permanent,” *IEEE Transactions on Aerospace and Electronic Systems* **53**(2), 698–702 (Apr. 2017).

11. B. Zhou and N. K. Bose, “Multitarget Tracking in Clutter: Fast Algorithms for Data Association,” *IEEE Transactions on Aerospace and Electronic Systems* **29**(2), 352–363 (Apr. 1993).

12. S. Maskell, M. Briers, and R. Wright, “Fast Mutual Exclusion,” Proceedings of the Proceedings of SPIE: Signal and Data Processing of Small Targets, volume 5428, Orlando, FL, 25 Aug. 2004, pp. 526–536.

13. P. Horridge and S. Maskell, “Real-Time Tracking of Hundreds of Targets with Efficient Exact JPDAF Implementation,” Proceedings of the Proceedings of the 9th International Conference on Information Fusion, Florence, Italy, 10–13 July 2006.

14. S. Maskell, “Signal Processing with Reduced Computational Complexity,” Feb. 2008.

15. “GB0315349.1 - Signal processing with reduced combinatorial complexity.” URL `https://www.ipo.gov.uk/p-ipsum/Case/ApplicationNumber/GB0315349.1`, Date Accessed: 12 July 2021.

16. J. Dezert and Y. Bar-Shalom, “Joint Probabilistic Data Association for Autonomous Navigation,” *IEEE Transactions on Aerospace and Electronic Systems* **29**(4), 1275–1286 (Oct. 1993).

17. D. F. Crouse, Y. Bar-Shalom, P. Willett, and L. Svensson, “The JPDAF in Practical Systems: Computation and Snake Oil,” Proceedings of the Proceedings of SPIE: Signal and Data Processing of Small targets, volume 7698, Orlando, FL, 15 Apr. 2010.

18. J. Dezert and Bar-Shalom, “Computational Complexity of the JPDA: Worst Case Analysis,” TN-2021-04-21, ONERA, Apr. 2021. URL `https://www.onera.fr/sites/default/files/297/OnJPDAComplexity-TechnicalNote2021-04-21.pdf`.

19. J. G. Michaels and K. H. Rosen, *Applications of Discrete Mathematics* (McGraw-Hill, Boston, 2007).

20. D. F. Crouse, "The Tracker Component Library: Free Routines for Rapid Prototyping," *IEEE Aerospace and Electronic Systems Magazine* **32**(5), 18–27 (May 2017).

21. D. F. Crouse, "The Tracker Component Library." URL `https://github.com/USNavalResearchLaboratory/TrackerComponentLibrary`, Accessed 12 July 2021.

22. D. F. Crouse, "On Implementing 2D Rectangular Assignment Algorithms," *IEEE Transactions on Aerospace and Electronic Systems* **52**(4), 1679–1696 (Aug. 2016).

23. H. J. Ryser, *Combinatorial Mathematics*, chapter 2, Number 14 in The Carus Mathematical Monographs (The Mathematical Association of America, 1963).

24. H. Minc, "Evaluation of Permanents," *Proceedings of the Edinburgh Mathematical Society* **22**(1), 27–32 (Feb. 1979).

25. A. Nijenhuis and H. S. Wilf, *Combinatorial Algorithms for Computers and Calculators*, 2 ed. (Academic press, New York, 1978).

26. A. Ben-Dor and S. Halevi, "Zero-One Permanent is #P-Complete, A Simpler Proof," Proceedings of the Proceedings of the 2nd Israel Symposium on Theory and Computing Systems, Natanya, Israel, 7–9 June 1993, pp. 108–117.

27. L. G. Valiant, "The Complexity of Computing the Permanent," *Theoretical Computer Science* **8**(2), 189–201 (1979).

28. B. Roos, "New Permanent Approximation Inequalitites Via Identities," *Lithuanian Mathematical Journal* **60**(2), 248–275 (Apr. 2020).