An Optimal Assignment Scheduler for Multifunction Phased Array Radars

Zhen Ding, Peter Moo and David DiFilippo

Radar Systems Section, DRDC Ottawa,
3701 Carling Avenue, Ottawa, ON, Canada K1A 0Z4

Keywords: Multi-function radar, optimal assignment scheduler, time balancing scheduler, accumulated delay time, maximum delay time, tracking occupancy, detection and tracking

Abstract

In this paper, we propose a multi-level Optimal Assignment Scheduler (OAS). The proposed scheduler is compared with a modified Time Balancing Scheduler (TBS). A scenario of 40 targets and over 400 detection beams is used for performance evaluation. All targets are updated every 1 to 2 seconds, which results in a varying number of tracking beams. The simulation results show that the OAS offers much better performance, with an accumulated delay time of only 1.62 seconds, comparing to 30.08 seconds of accumulated delay time by TBS. The maximum delay times are 0.06 second and 0.49 second for OAS and TBS, respectively. An additional benefit of OAS is that it could incorporate task priority information into the cost matrix for more desirable scheduling assignment.

1. Introduction

In a recent survey, it was noted that current phased array radar resource management algorithms can be categorized into six areas: including artificial intelligence algorithms, dynamic programming algorithms, Q-RAM algorithms, waveform aided algorithms, adaptive update rate algorithms and the US NRL benchmark algorithms [5]. An MFR typically has two sets of priorities; namely function priorities and task priorities. The function priorities are pre-determined by the radar mission. Task priorities are relatively numbered within the same function. For example, tracking tasks may have priority values from 0 to 1 [7]. Track priority has been used to determine track update rate [10]. In this study, we focus on task scheduling, assuming track priorities are available.

In a recent survey, phased array radar resource management algorithms are categorized into six areas, including artificial intelligence algorithms, dynamic programming algorithms, Q-RAM algorithms, waveform aided algorithms, adaptive update rate algorithms and the US NRL benchmark algorithms [5]. A general MFR resource management system performs the following steps:

- Get a radar mission profile and determine function priority sequence;
- Generate radar tasks;
- Assign priorities to tasks by using a prioritization algorithm;
- Manage available resources by a scheduling algorithm so that the system can meet the requirements of all radar functions;
- When there are no detections in the course of non-surveillance tasks, a re-look may be scheduled based on its priority and elapsed time since the last scheduling of the same task;
- The radar scheduler considers radar beams, dwell time, carrier frequency, PRF, energy level, etc.

An MFR typically has two sets of priorities: function priorities and task priorities. The function priorities are pre-determined by the radar mission. For example, the following priority levels exist:

1. High priority tracks (highest priority)
2. Track maintenance
3. Medium priority tracks
4. Plot confirmation
5. Track initiation
6. Low priority tracks
7. Surveillance and slow tracks
8. Receiver calibration and built-in-test (BIT) (lowest priority)

Task scheduling can be interleaving and non-interleaving. In a few references, interleaving algorithms are proposed [6, 11], where there is an idle period between a pulse transmission and its receiving. The interleaving approach is not very practical. In this study, we use a model that the sending and receiving sub-tasks are considered as one unique task.

Most radar scheduling algorithms fall into non-interleaving category. For example, Winter introduced a local search method to compute efficient schedules [13]. Cost functions for data link, tracking and searching are formulated. Linear programming is used to find the optimal schedule. Moo proposed a Sequential Scheduler [9]. The Two-Slope Benefit Function (TSBF) Sub-Scheduler is used and requires that each tracking look and high-priority surveillance look has a benefit function, which specifies benefit as a function of start time. The Time Balancing Scheduler (TBS) was a simple and efficient algorithm [2, 12].

In particular, the TBS was originally proposed and implemented in the MESAR system [2, 4, 12]. The scheduler keeps a time balance for each function. At any scheduling time, the radar picks the function with the maximum time balance for scheduling.

In this paper, we propose a multi-level Optimal Assignment Scheduler (OAS), which is tested against the TBS. Accumulated scheduling delay and maximum delay are used as the performance measures. A simulation with over 400 detection beams and 40 targets is used to compare the two schedulers. In the simulation, each target requests a random update interval of 1to 2 seconds.

The rest of the paper is arranged as follows. The new scheduling algorithms are presented in Section 2. The benchmark scheduler (TBS) is described in Section 3. A simulation scenario is given in Section 4. The performance evaluation is provided in Section 5 and the last section concludes the paper.

2. Optimal Assignment Scheduler

Assume the radar has L functions and a time window $[t_k, t_{k+1}]$ is considered for scheduling. Within the scheduling time window, the radar is requested $[n_1, \cdots, n_L]$ beams by the L functions, respectively. The OAS includes four basic steps. A diagram of the scheduler is shown is Figure 1.

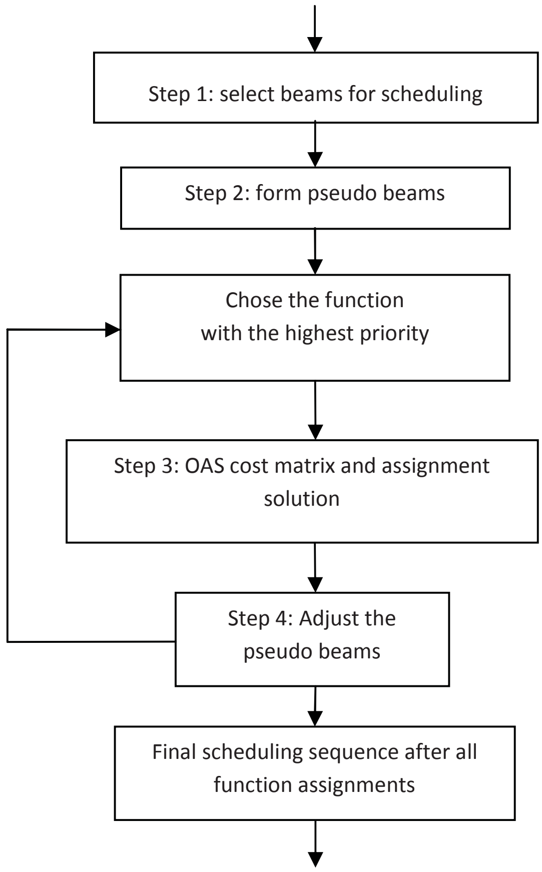

Figure 1: A diagram of OAS.

1. Selection of beams: this process is mission strategic and determines how many beams will be scheduled. In an over-loading situation, some tasks will be discarded. For example, if the mission profile specifies the maximum tracking occupancy, the radar will only able to schedule a maximum number of tracking beams within a frame time.

2. Formation of pseudo beams: this is created for all selected beams. The pseudo beams have time stamps, by simply using the detection dwell time in our proposed algorithm. No beam positions in azimuth and elevation are assigned in this step.

3. Scheduling by OAS: this step finds the unassigned function with the highest priority so that the beams are assigned for a higher priority function first. A cost matrix $[C_{ij}]$ can be built based on the task priority and time difference between the task time and the pseudo beam time:

$$[C_{ij}] = f(\Delta t_{ij}, p_i)$$

where Δt_{ij} is the time difference between task i and pseudo beam j. p_i is the priority of task i. The simplest cost function is the time difference where all tasks have the same priority.

$$[C_{ij}] = \Delta t_{ij}$$

If tasks have different priorities, the cost can be the product of the time difference and the priority value.

$$[C_{ij}] = \Delta t_{ij} * p_i$$

4. Revise the pseudo beam list by replacing the assigned positions for the tasks in the function which is currently considered. Repeat step 3 until the last function, typically the surveillance, which will be assigned to the unassigned beams.

3. Time Balancing Scheduler

The TBS is a method that is often used in operating systems to dynamically allocate times for different processes. Figure 2 shows a time balancing graph, where two functions have their own time balances. The slope and drop are the two parameters needed for the time balance of each functions. For convenience, the slope is set to 1. By adjusting the slope values, a desirable tracking occupancy can be achieved. For example, in order to achieve 20% occupancy, the drops for the detection and tracking are 1 and 4, which means there is 1 tracking beam after 4 detection beams. For example, a scheduling sequence is:

[1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0]

where “1” stands for a tracking beam and “0” stands for a detection beam. The occupancies are 4/5, 1/5 for the two functions.

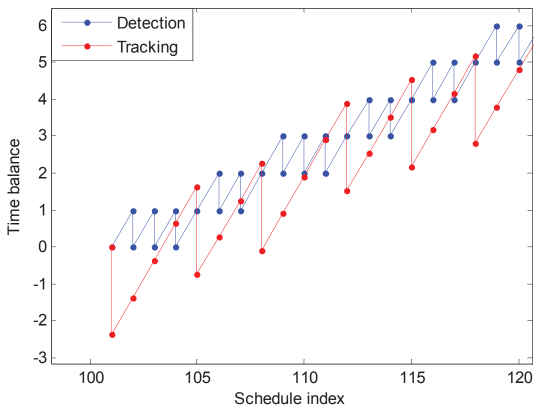

Figure 2: A TBS graph for 20% tracking occupancy.

When more than two functions are involved, it becomes more difficult to determine the occupancy. For instance, tracking 2 is prior to tracking 1, resulting in the following schedules:

[2,1,0,0,0,0,2,1,0,0,0,0,2,1,0,0,0,0,2,1]

where “2” represents tracking 2. The occupancies are: 4/6, 1/6, 1/6 for the three functions.

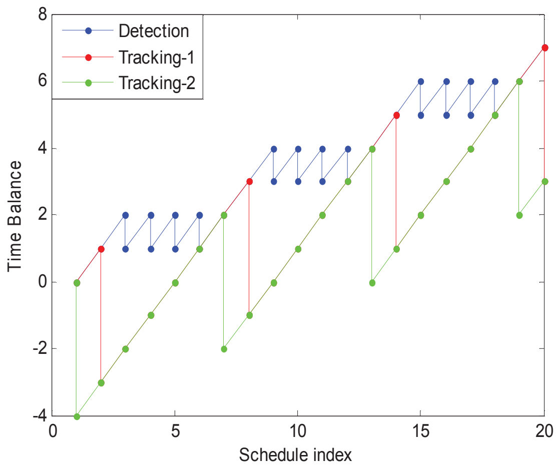

Figure 3: A TBS graph of three functions.

The scenario is passed to a selection process, which determines how many tracking beams are to be scheduled. In the example in Figure 3, assuming the maximum track occupancy of 30%, all tracking beams will be scheduled. Note that the maximum tracking occupancy is a strategic decision, not an algorithmic decision. Once the maximum tracking occupancy is given, the maximum radar frame time can be calculated. Often, the maximum frame time is a surveillance requirement, which determines how fast a surveillance area can be revisited by the radar search beams.

4. A Simulation Scenario

For simplicity, we consider phased array radar with two functions: surveillance and tracking. The surveillance has a fixed number of 469 beams, each having a particular azimuth and elevation. There are 40 targets with confirmed tracks. Each track requests an update interval of 1 to 2 seconds. The dwell time for both surveillance and tracking is 0.01 second. The time window for assignment is [0, 6.25] seconds. Within this period of time, 156 track beams are requested. This requires a tracking occupancy of 156/(156+469) = 25%.

The scenario is passed to a selection process, which determines how many tracking beams are to be scheduled. Assuming the maximum track occupancy of 30%, all tracking beams will be scheduled. Note that the maximum tracking occupancy is a strategic decision, not an algorithmic decision. Once the maximum tracking occupancy is given, the maximum radar frame time can be calculated. Figures 5 and 6 show all the requested beams. It can be seen that multiple beams are required for the same time, resulting in many scheduling conflicts. For illustration purpose, the beam requests of the first 10 targets are listed in Table 1.

Table 1: Beam requests of the first 10 targets

Target index	Update requests (seconds)			
1	1	2.6	3.8	5.3
2	1.4	2.9	4.3	5.4
3	1.8	3.6	4.8	6
4	1.6	3	4.2	6.1
5	1.1	2.2	3.3	4.9
6	1	2.9	4.6	/
7	1	2.8	4.7	/
8	1.8	3.5	5	6.1
9	1.1	2.1	4	5.3
10	1.3	2.7	4.3	5.3

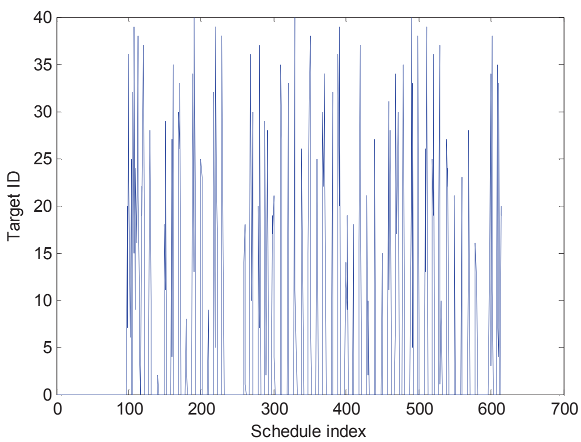

Figure 4: All targets and their requested scheduling times.

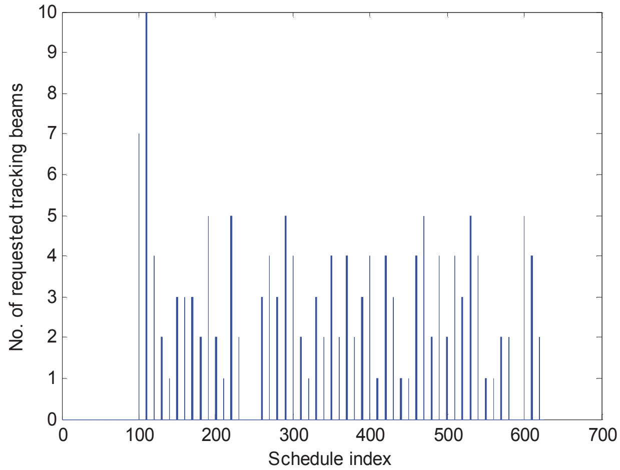

Figure 5: Requested 156 tracking beams.

5. Performance Evaluation

The TBS and OAS are specified as follows:

1. The whole frame is used as the scheduling time window;

2. Tracking function has a higher priority over the detection function;

3. All tracking tasks have equal priority and the time difference is used as the cost function in OAS;

4. Both schedulers are designed for two-level assignment. First, tracking beams are assigned. For OAS, the assignment matrix is set up and the Auction algorithm is used to find the best solution [11].

5. The TBS is modified to handle the same two-level assignment. Since only two functions are considered at each level, the occupancy can be exactly implemented. The tracking beams start at 1 second when the TBS starts. The drop values are 1 and 2.3654 for detection and tracking, respectively (see Figure 6). The value 2.3654 was calculated by 156/525, the ratio of the number of tracking beams and the number of total beams within the time balance window. All the tracking beams generated are assigned to the

tracking beams based on the requested scheduling time, the earlier the first.

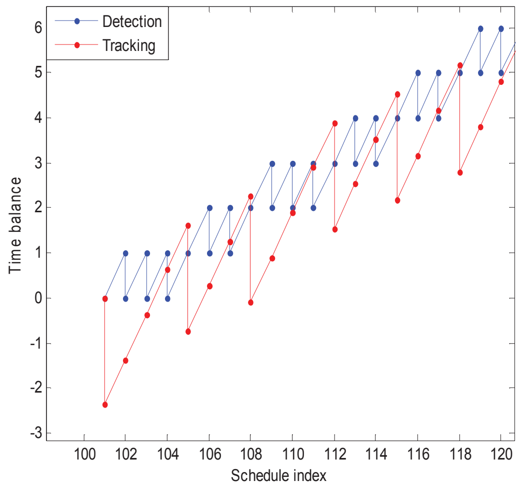

Figure 6: Part of the graph of the TBS.

For OAS, the assignment matrix is set up and the Auction algorithm is used to find the best solution [1]. The testing scenario was processed by the two schedulers. The two performance measures used are the maximum delay and the accumulated delay. The results are shown in Table 1 and Figures 7 and 8. OAS has much better performance. Figure 9 compares the planned beams with 20% occupancy, the requested tracking beams and the actual scheduled tracking beams, where the requests with the same time are expanded near the request.

Table 1: Performance of two schedulers.

	TBS	OAS
maximum delay	0.49 s	0.06 s
accumulated delay	30.08 s	1.62 s

The TBS offers a uniform interval between tracking beams, which offsets the scheduling time from the requested time of tracking tasks. The OAS attempts minimize the total offset, i.e., the summation of all individual offsets.

In the simulation, we used one whole frame time to optimize the assignment. In practice, it is beneficial to have a much shorter scheduling time window, such as 0.5 to 1 seconds. With a shorter window, the radar is able to insert additional urgent beams. In addition, the cost matrix has smaller dimensions, which significantly improves the efficiency of the optimization algorithm.

Also notice that we used the search dwell to form the initial pseudo beams. This formulation is accurate when the tracking waveform and search waveform have the same dwell time. If the dwell times are different, the formulation becomes an approximate one. The general formulation for the assignment cost matrix will be investigated in future work. We would also like to find out how the scheduling delays affect the final tracking performance, one of the ultimate goals of the radar resource management.

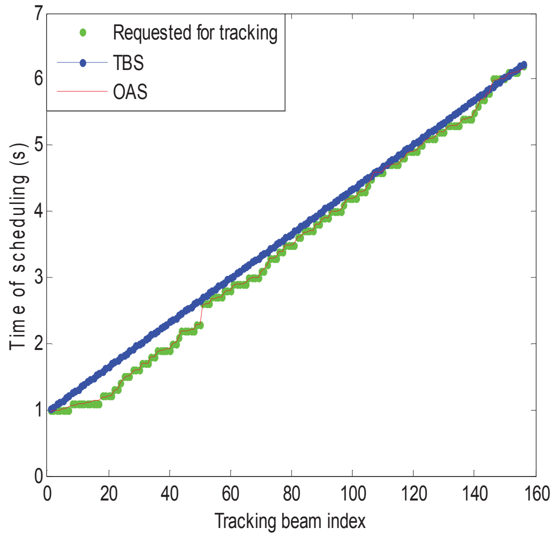

Figure 7: Comparison of scheduling time.

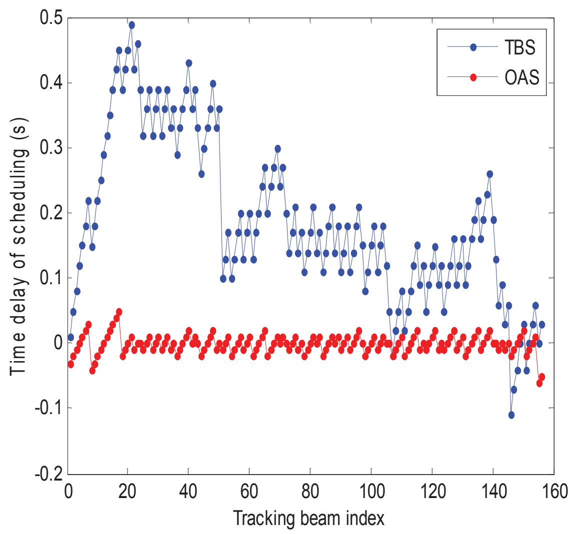

Figure 8: Comparison of maximum delays.

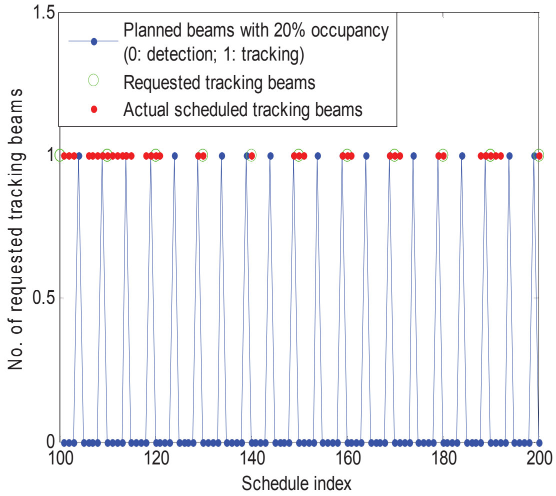

Figure 9: Scheduling results of OAS within 1-2 seconds.

6. Conclusion

A new radar scheduler named OAS was proposed based on formulation of the radar scheduling into an optimal assignment problem. Using simulated radar data with detection and tracking tasks, the proposed OAS was tested and compared with an existing scheduler, the TBS. The new scheduler has much better performance as quantified in terms of the accumulated delay and maximum delay. In addition, the cost matrix formulation allows the use of priority information of each task within the same function, overcoming a drawback of other schedulers.

References

[1] Bertsekas, D. P., "An auction algorithm for shortest paths", *SIAM Journal on Optimization*, 1, pp. 425-447, 1991.

[2] Butler, J.M., Moore, A. R. and Griffiths, H.D., “Resource management for a rotating MFR”, In *Proceedings of the IEEE International Radar Conference*, pp. 568-572, 1997.

[3] Butler, J. M., “Tracking and control in multi-function radar”. *Ph.D. thesis*, University College London, 1998.

[4] Capraro, G. T., Farina, A., Griffiths, H. and Wicks, M. C., “Knowledge-based radar signal and data processing”, *IEEE Sig. Proc. Mag.*, pp. 18–29, 2006.

[5] Ding, Z., “A survey of radar resource management algorithms”, In *Proceedings of the 11th Canadian Conference on Electrical and Computer Engineering*, pp. 1559-1564, Niagara Falls, 2008.

[6] Duron, C. and Proth, J. M., “Insertion of a random bitask in a schedule: a real-time approach”, *Computers & Operations Research*, 31, pp. 779-790, 2004.

[7] Miranda, S. L. C., Baker, C. J., Woodbridge K. and Griffiths, H.D., “Fuzzy logic approach for prioritisation of radar tasks and sectors of surveillance in multifunction radar”, *IET Radar Sonar & Navigation*, 1 (2), pp. 131–141, 2007.

[8] Miranda, S. L. C., Baker, C., Woodbridge, K. and Griffiths, H. D., “Multifunction radar resource management”, Chapter 10 in “*Knowledge-based radar detection, tracking, and classification*”, Edited by Fulvio Gini and Muralidhar Rangaswamy, Wiley-Interscience, 2008.

[9] Moo, P., “Scheduling for multifunction radar via two-slope benefit functions”, *IET Radar, Sonar & Navigation,* 5 (8), pp. 884-894, 2011.

[10] Noyes, S. P., “Calculation of next time for track update in the MESAR phased array radar”, *(Digest No. 1998/282), IEE Colloquium on Target Tracking and Data Fusion*, pp. 1-7, June 1998.

[11] Orman, A. J., Potts, C.N., Shahani, A. K. and Moore A. R., “Scheduling for a multifunction phased array radar system”, *European Journal of Operational Research*, 90 (1), pp. 13-25, 1996.

[12] Stafford, W. K., “Real time control of multifunction electronically scanned adaptive radar”. *IEE Colloquium on Real Time Management of Adaptive Radar Systems*, 1990.

[13] Winter, E. and Lupinski, L., “On scheduling the dwells of a multifunction radar”, *International Conference on Radar*, Shanghai, Oct. 2006.