A Heuristic Task Scheduling Method for Multifunction Radar

H.Mir[1] and F. Ben Abdelaziz[2]
[1]*Department of Electrical Engineering, American University of Sharjah, Sharjah, UAE.*
[2]*Department of Information Systems, Supply Chain Management and Decisions, Neoma Business School, Rouen, France*
hmir@aus.edu

***Abstract*— Modern radar systems are required to perform various tasks within a given time window in order to ascertain the presence of a new target or update information about an existing target. The scheduling of such tasks is therefore important in order to efficiently utilize the radar timeline. This paper describes a novel heuristic approach for scheduling tasks on a multifunction radar. The proposed approach is based on tabu search, and computational results are presented to assess the efficacy of the proposed method.**

***Index Terms*— Multifunction Radar; Task Scheduling.**

I. INTRODUCTION

The operation of a radar can be described as a sequence of operations (which will be termed as a *task*) whereby a signal is transmitted for an interval of time, following which the radar is idle for an interval of time that is governed by the minimum target range of interest, and finally a receive interval in which the radar records any signals that have been reflected from potential targets. A multifunction radar can perform tasks in which the duration of the respective transmit, wait, and receive interval may vary, depending on the nature of task. For example, an air defense task will typically have a much shorter wait interval compared to a missile defense task. Given a set of tasks, each of which as its own unique values for the duration of the transmit, wait, and receive interval, the radar task scheduling problem involves determining the ordering of the tasks on the timeline so that their execution is completed before a specified deadline.

Previous work on radar task scheduling has involved a variety of approaches such as nonlinear control [1], fuzzy logic [2], and optimization models [3]-[7]. The cyclic variant of the radar task scheduling problem is considered in [8-11]. This paper summarizes a heuristic for the radar task scheduling problem that is based on tabu search [12]. The proposed method is distinguished by its explicit modeling of the components of a radar task (as opposed to many existing approaches which compound the transmit, wait, and receive period into a single dwell) that can occur along an unstructured timeline. Moreover, the proposed heuristic automatically performs task compaction by exploiting the idle wait time of one task in order to commence the transmit/receive period of another task.

II. PROPOSED RADAR TASK SCHEDULING METHOD

Figures 1-3 show the flow diagram of the proposed radar task scheduling method. An overview of the proposed radar task scheduling method is shown in Figure 1. A list $\mathcal{X}$ is created that contains task start times sorted in increasing order, while another list $\mathcal{S}$ is created that contains the corresponding task indices. Together, $(\mathcal{S}, \mathcal{X})$ describe a task schedule. An initial schedule is first generated at random. However, any candidate schedule $(\mathcal{S}, \mathcal{X})$ is only valid if it satisfies a feasibility check. In particular,

- All executed tasks must terminate before the deadline H.
- The transmit and receive sub-tasks must not collide.

These checks are formalized in Figure 2. Note that T_n^1, T_n^2, and T_n^3 denote the duration of the transmit, wait, and receive interval, respectively, of task n.

As shown in Figure 1, after a feasible initial schedule has been generated, $\mathcal{S}$ and $\mathcal{X}$ are added to the corresponding "tabu" lists $\mathcal{T}_S$ and $\mathcal{T}_X$. The term "tabu" is used because the schedules in the tabu lists are not used as a performance benchmark. The tabu search proceeds by generating neighboring schedules $(\mathcal{S}_g,\mathcal{X}_g)$ of the schedule $(\mathcal{S},\mathcal{X})$ through four possible operations:

- A randomly selected task in $\mathcal{S}$ is deleted.
- A randomly selected task not in $\mathcal{S}$ is inserted at a random position in $\mathcal{S}$.
- A randomly selected task not in $\mathcal{S}$ replaces a task at a random position in $\mathcal{S}$.
- The position of two randomly selected tasks in $\mathcal{S}$ is swapped.

This process is formalized in Figure 3. Each generated neighboring schedule is subjected to the same feasibility check in Figure 2.

Returning to Figure 1, the best neighboring schedule is selected as the one that schedules the most tasks, and the number of tasks scheduled is stored in the variable J_{NEW} and compared to the value of J_{OLD} from the best neighbor schedule of the previous iteration. The algorithm terminates if a specified number of iterations N_{iter} yield no improvement in the number of scheduled tasks or if all tasks have been scheduled.

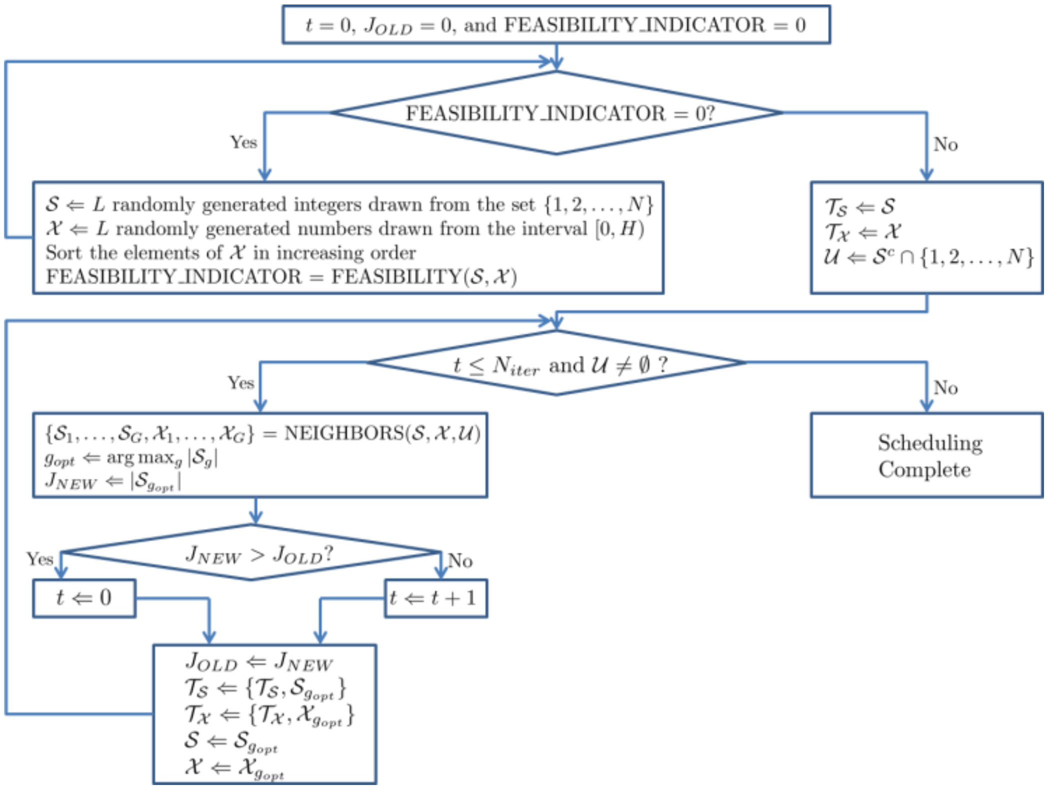

Figure 1: Flow Diagram Overview of Proposed Radar Task Scheduling Method.

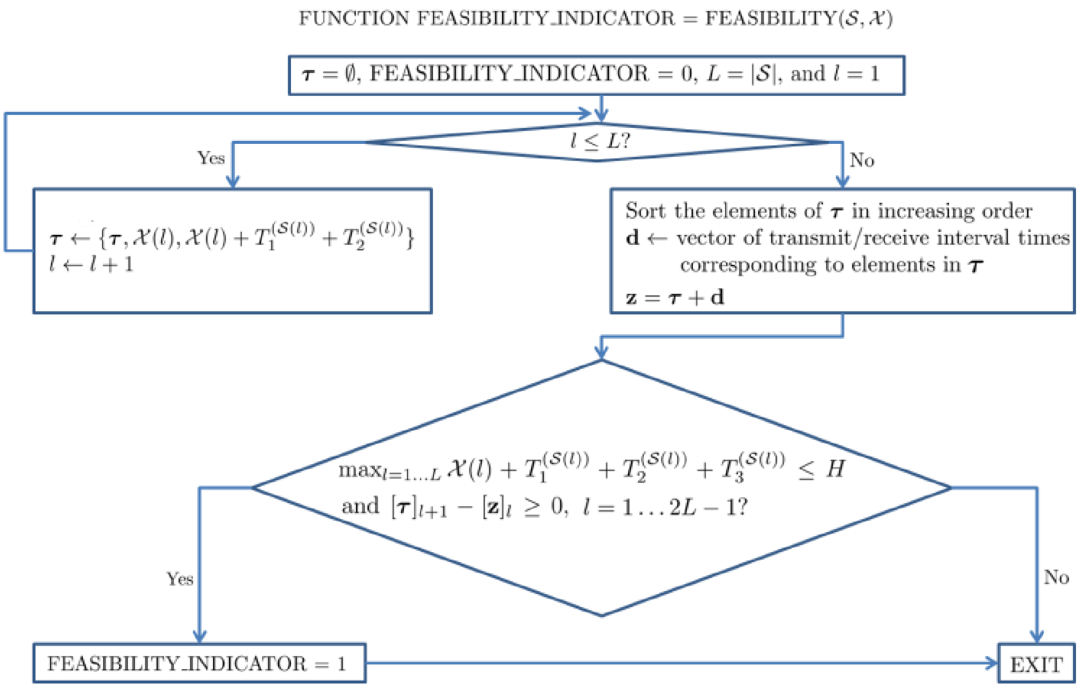

Figure 2: Flow Diagram For Feasibility Check of Candidate Schedule.

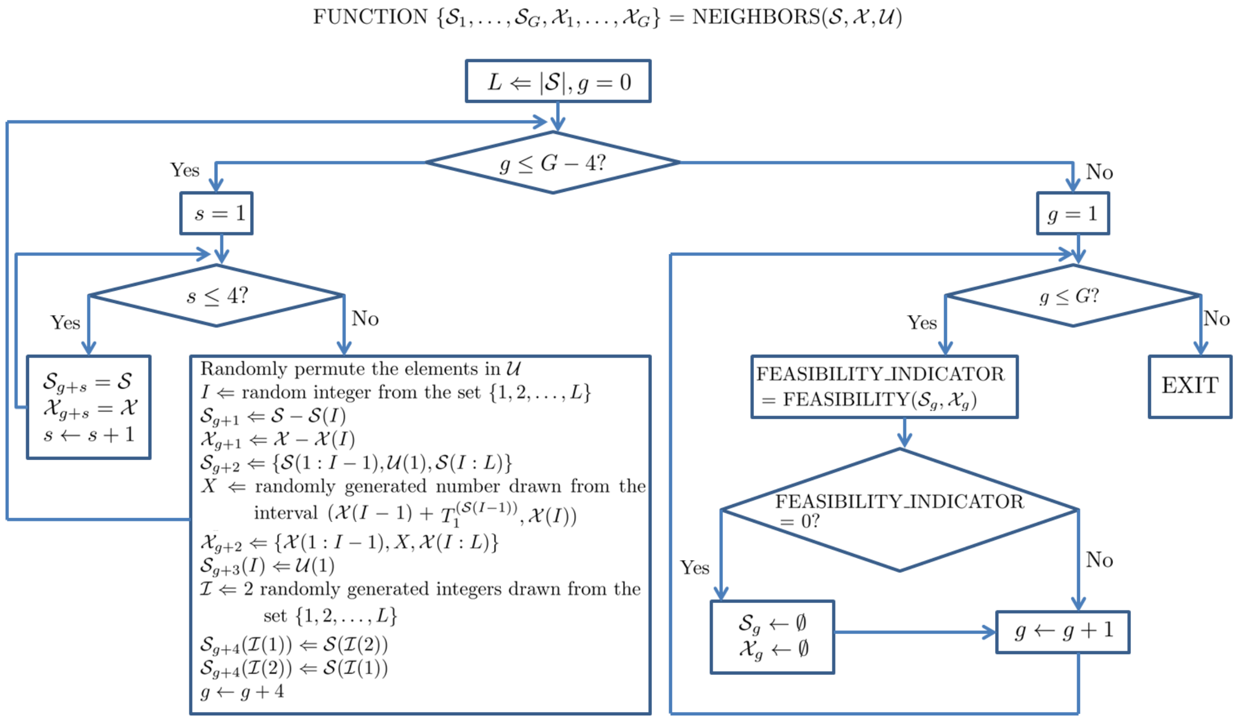

Figure 3: Flow Diagram for Generation of Neighboring Solutions.

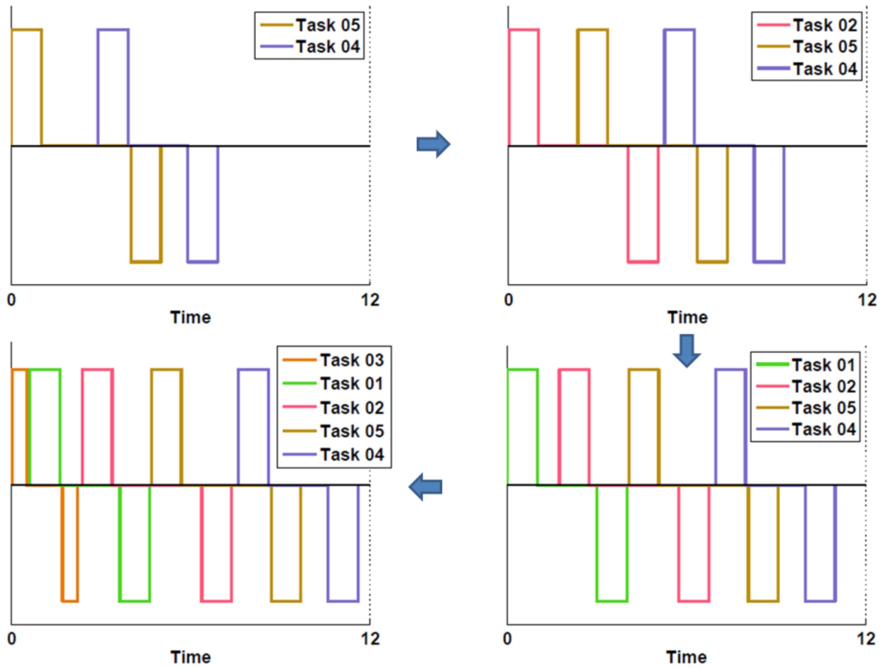

Figure 4: Schedule Evolution Using Proposed Heuristic Scheduling Method for Tasks In Table 1.

III. Computational Results

Consider first a small scheduling example consisting of 5 tasks with parameters as shown in Table 1 and a deadline of $H = 12$. Note that naive sequential placement of tasks results in an execution time of 20, which exceeds the deadline. The proposed scheduling method, however, automatically performs task compaction by exploiting the wait interval of a task to commence the transmit/receive interval of another task. The evolution of the schedule using the proposed heuristic method is shown in Figure 4, from which it can be seen that ultimately all tasks are successfully scheduled.

Table 1
Example Task Parameters

Task Number	Transmit Duration	Wait Duration	Receive Duration
1	1	2	1
2	1	3	1
3	0.5	1.2	0.5
4	1	2	1
5	1	3	1

Next, a large-scale simulation is performed using task parameters based on those in [13]. The deadline is set to H = 200ms. The performance of the proposed heuristic method is assessed in terms of the percentage of dropped tasks, and is benchmarked against a sequential heuristic which performs task compaction and scheduling in two distinct stages ([3],[10]). The results are shown in Figure 5. The improved performance of the proposed heuristic can be attributed to the integrated scheduling and task compaction behavior. This results in tasks being scheduled only as needed and leaves more room on the timeline to accommodate new tasks. In contrast, the sequential task heuristic performs task compaction first; while this compaction may result in a shorter execution time, it also causes some tasks with a lower update rate to be unnecessarily repeated, thus reducing the room on the timeline for new tasks and causing more tasks to be dropped. Figure 6 compares the run times of the proposed heuristic and sequential heuristic. The proposed heuristic is indeed somewhat slower than the sequential heuristic; however, as Figure 5 shows, the modest increase in run time of the proposed heuristic results in a significant reduction of the percentage of dropped tasks over the sequential heuristic.

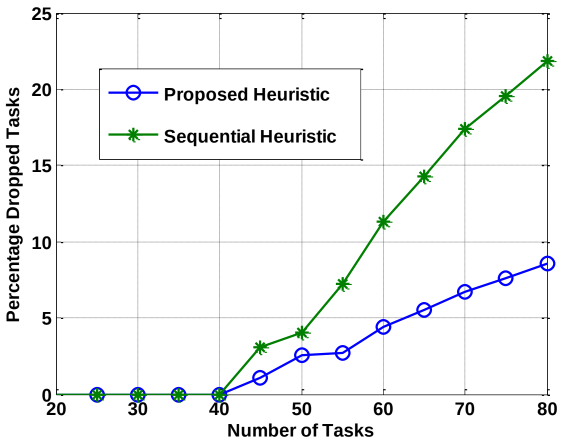

Figure 5: Comparison of Percentage of Dropped Tasks For Proposed And Sequential Heuristic Scheduling Method.

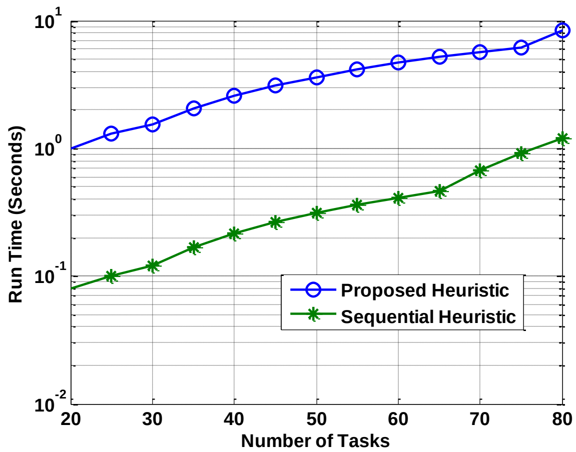

Figure 6: Comparison of Run Times for Proposed and Sequential Heuristic Scheduling Method

IV. Conclusion

A radar task scheduling heuristic based on tabu search was proposed. The proposed heuristic elegantly integrates task compaction and scheduling. Compared with previously published heuristics that treat task compaction and scheduling as separate steps, the proposed heuristic was demonstrated to exhibit improved utilization of the radar timeline, thus resulting in fewer dropped tasks.

References

[1] S-M. Hong and Y-H. Jung, "Optimal scheduling of track updates in phased array radars," *IEEE Transactions on Aerospace and Electronic Systems*, vol.34, pp. 1016-1022, Jul. 1998.

[2] S. Miranda, "Knowledge-based resource management for multifunction radar: a look at scheduling and task prioritization," *IEEE Signal Processing Magazine*, pp. 66-76, Jan. 2006.

[3] S. Ghosh, R. Rajkumar, J. Hansen, and J. Lehoczky, "Integrated QoS-aware resource management and scheduling with multi-resource constraints," *Real-Time Systems*, pp. 7-46, Jul. 2006.

[4] E. Winter and P. Baptiste, "On scheduling a multifunction radar," *Aerospace Science and Technology*, pp. 289-284, May 2007.

[5] Q. Zheng, F. Barbaresco, and P.Baptiste, "On scheduling a multifunction radar with duty cycle budget," *2009 Cognitive Systems with Interactive Sensors*, pp. 1-6, Nov. 2009.

[6] P. Baptiste and R. Sadykov, "Time-indexed formulations for scheduling chains on a single machine: An application to airborne radars," *European Journal of Operations Research*, pp. 476-483, Jun. 2010.

[7] V. Jeauneau, T. Guenais, and F. Barbaresco "Scheduling on a fixed multifunction radar antenna with hard time constraint," *2013 International Radar Symposium*, pp. 1-6, Jun. 2013.

[8] H. Mir and J.Wilkinson, "Task scheduling algorithm for an air and missile defense radar," *2008 IEEE Radar Conference*, pp. 1629-1634, May 2008.

[9] H. Mir and F. Abdelaziz, "Cyclic task scheduling for multifunction radar," *IEEE Transactions on Automation Science and Engineering*, pp. 529 - 537, Jul. 2012.

[10] H. Mir and A. Guitouni, "Variable dwell time task scheduling for multifunction radar," *IEEE Transactions on Automation Science and Engineering*, pp. 463 - 472 , Apr. 2014.

[11] Z. Spasojevic, S. Dedeo and R. Jensen, "Dwell scheduling algorithms for phased array antenna," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 42-54, Jan. 2013.

[12] F. Abdelaziz and H. Mir, "An optimization model and tabu search heuristic for scheduling of tasks on a radar sensor," *IEEE Sensors Journal*, vol. 16, pp. 6694 - 6702, Sep. 2016

[13] T-W. Kuo et. al., "Real-time subtask scheduling of component-oriented phased array radars," *IEEE Transactions on Computers*, vol. 54, pp. 47-60, Jan. 2005.