

A Near-optimal Slot Assignment Algorithm for RFID Reader Networks

Chun-Fu Lin¹, Frank Yeong-Sung Lin¹, Cheng-Ta Lee^{1,2}

Department of Information Management, ¹National Taiwan University (NTU), ²Lan Yang Institute of Technology
No.1, Sec. 4, Roosevelt Road, Taipei, 106 Taiwan, R.O.C.
{d92002, yslin, d90001}@im.ntu.edu.tw

Abstract

In this paper, we propose a method that reduces the cycle time of RFID reader networks by overlapping slots. A mathematical formulation of the problem is specified and an effective heuristic algorithm is developed.

Keywords: Anti-Collision Problems, Lagrangean Relaxation, reader network, RFID

1. Introduction

In technologies that use radio waves, RFID (Radio Frequency Identification) is considered an effective alternative of object tracking and identification. An RF scan can be performed at a distance, and the data can be read without line-of-sight. Moreover, compared to traditional bar codes, the RF tags are more resistant to environmental damage. RFID applications range from object tracking and identification (e.g., supply-chain management) to environmental sensing (SRFID, e.g., WISP [7]). The effective deployment of RFID reader networks is crucial to satisfying management constraints, as incorrect deployment could result in the reader collision problem [2][3] which is regarded as NP-complete. To avoid collisions, a reader is scheduled to access tags at a different time to its neighbors.

“Colorwave” [3] is a distributed algorithm that solves the reader collision problem. The basic concept is that every reader chooses a random color (slot number) independently, within a cycle defined by the MAX_COLOR component. The selection (referred to as “kick”) is sent out to inform the reader’s neighbors and force them to change their colors, if they are the same as that of the active reader. The Colorwave algorithm implements a dynamic mechanism by adjusting MAX_COLORS up or down according to the number of the successful transmissions. If two adjacent readers overlap but they are not close enough to communicate with each other, the neighbor may not receive the “kick”, which could cause oscillations.

If there are a large number of tags in a reader’s interrogation zone, the reader should select a small MAX_COLOR value, which represents a short cycle time. However, if there are many neighbors or the collision rate is high, a small MAX_COLOR value is

unobtainable. This raises a concern about fairness, as a reader with more tags should either be allocated more slots in a cycle, or be given a shorter cycle.

In the EPC network [4], readers connect to servants (middleware), which perform data aggregation, filtering, and reporting. The connection to a backhaul can be made directly via a wireless or plug. Then, with the wireless and ad-hoc capabilities, the readers can be accessed by a server without physical links, and the slot assignment can be scheduled. The server periodically calculates and sends out the arrangement to refresh all readers by monitoring the network’s status so that the impact of collisions is minimized and the system’s constraints are satisfied. In a global cycle arrangement, satisfying every reader’s requests would result in a large cycle time. Thus, to reduce the cycle, we consider allowing readers to overlap as long as there are other slots available to the active reader. For example, all commodities on a smart-shelf within one reader’s interrogation zone might remain unchanged, such that to keep sending identical data would be meaningless. In such a case, the reader may decide to slow down its reading rate and allow some of its slots to be used by its neighbors as long as there are other slots available for its own use.

The mathematical model proposed in [1] for the channel assignment problem [8] can be extended to solve this problem. We consider overlapped readers to be durable under the system’s constraints and propose a general mathematical framework to model the problem.

2. Problem Description

Given a set of readers $L = \{1, 2, \dots, n\}$, a set of available time slots $F = \{1, 2, \dots, f\}$, and a GPS capability or a pre-defined deployment, we can calculate an adjacency matrix A_{jk} . We formulate the problem as an integer programming problem, where the objective function is to minimize the number of assigned slots (cycle length) subject to some constraints. The notations used in this paper are listed in Table 2.1 and 2.2.

Table 2.1 Notations for the given parameters

Notation	Description
L	Set of readers

F	Set of time slots
c_j	Slot requirement of reader j
α	Reader-wise threshold. The number of overlapped slots divided by the number of slots assigned to each reader must not be larger than this value.
β	System-wise threshold. The sum of overlapped slots of all readers divided by the sum of assigned slots of all readers must not be larger than this value.
A_{jk}	The adjacency matrix of readers. If the interrogation zones of reader j and reader k overlap, then A_{jk} will be set to 1; otherwise 0.

Table 2.2 Notation descriptions for decision variables

Notation	Description
B_{ij}	The overlapped slot matrix. If slot i assigned to reader j is also assigned to any of j's neighbors, B_{ij} is set to 1; otherwise 0.
x_{ij}	If slot i is assigned to reader j, x_{ij} is set to 1; otherwise 0
y_i	The slot assignment array. If slot i is assigned to any reader, y_i is set to 1; otherwise 0.

Objective function:

$$Z_{IP} = \min \sum_{i \in F} y_i,$$

subject to:

$$\sum_{i \in F} B_{ij} \leq \alpha \sum_{i \in F} x_{ij} \quad \forall j \in L \quad (1)$$

$$\sum_{j \in L} \sum_{i \in F} B_{ij} \leq \beta n \sum_{i \in F} y_i \quad (2)$$

$$(x_{ij} + x_{ik})A_{jk} \leq B_{ij} + A_{jk} \quad \forall i \in F, \forall j \in L, \forall k \in L \quad (3)$$

$$c_j \leq \sum_{i \in F} x_{ij} \quad \forall j \in L \quad (4)$$

$$x_{ij} \leq y_i \quad \forall i \in F, \forall j \in L \quad (5)$$

$$x_{ij} = 0 \text{ or } 1 \quad \forall i \in F, \forall j \in L \quad (6)$$

$$B_{ij} = 0 \text{ or } 1 \quad \forall i \in F, \forall j \in L \quad (7)$$

$$y_i = 0 \text{ or } 1 \quad \forall i \in F \quad (8)$$

Constraint (1) ensures that the ratio of the number of the overlapped slots and the number of assigned slots of a reader must be smaller than α . An extreme case is $\alpha = 0$ and $\alpha = 1$, which respectively indicate that there is no overlap and all assigned slots must be overlapped in one reader. Constraint (2), which is similar to constraint (1), ensures that the ratio of the number of overlapped slots and the number of assigned slots of all readers are smaller than β . Constraint (3) ensures that the value of B_{ij} is set to 1 if slot i, which is assigned to reader j, is also assigned to any neighbor of reader i; otherwise, it is set to 0. Constraint (4) requires that the slot assignment must satisfy the slot requirement

of every reader. Constraint (5) ensures that when slot i is assigned to any reader j, then y_i is set to 1. Constraints (6), (7) and (8) are integer constraints for the decision variables x_{ij} , B_{ij} , and y_i respectively.

3. Solution Procedure

First, to solve the problem Z_{IP} , we use the Lagrangean relaxation approach [5] [6], which comprises the following two steps: (1) multiply some complicating constraints by their corresponding Lagrangean multipliers, and (2) add the modified constraints to the primal objective function. According to the Lagrangean relaxation procedures, we can transform the primal problem into a Lagrangean relaxation problem (LR). In our case, we choose to dualize the Constraints (1), (2), (3) and (5) as follows:

$$\begin{aligned} Z_D(s, t, u, v) = & \min \left\{ \sum_{i \in F} y_i \right. \\ & + \sum_{j \in L} s_j \left(\sum_{i \in F} B_{ij} - \alpha \sum_{i \in F} x_{ij} \right) \\ & + t \left(\sum_{i \in F} \sum_{j \in L} B_{ij} - n\beta \sum_{i \in F} y_i \right) \\ & + \sum_{i \in F} \sum_{j \in L} \sum_{k \in L} u_{ijk} [(x_{ij} + x_{ik})A_{jk} - (B_{ij} + A_{jk})] \\ & \left. + \sum_{i \in F} \sum_{j \in L} v_{ij} (x_{ij} - y_i) \right\} \\ = & \min \left\{ \sum_{i \in F} (1 - tn\beta - \sum_{j \in L} v_{ij}) y_i \right. \\ & + \sum_{i \in F} \sum_{j \in L} (s_j + t - \sum_{k \in L} u_{ijk}) B_{ij} \\ & + \sum_{i \in F} \sum_{j \in L} (2 \sum_{k \in L} u_{ijk} A_{jk} - \alpha s_j + v_{ij}) x_{ij} \\ & \left. + \sum_{i \in F} \sum_{j \in L} \sum_{k \in L} u_{ijk} (-A_{jk}) \right\} (LR), \end{aligned}$$

subject to Constraints (4), (6), (7), and (8).

(LR) can be further decomposed into four independent solvable sub-problems as follows:

Sub-problem 1: (related to decision variable y_i)

Objective function:

$$Z_{sub1} = \min \sum_{i \in F} (1 - tn\beta - \sum_{j \in L} v_{ij}) y_i,$$

subject to (8).

Sub-problem 2: (related to decision variable B_{ij})

Objective function:

$$Z_{sub2} = \min \sum_{i \in F} \sum_{j \in L} (s_j + t - \sum_{k \in L} u_{ijk}) B_{ij},$$

subject to (7).

Sub-problem 3: (related to decision variable x_{ij})

Objective function:

$$Z_{sub3} = \min \sum_{i \in F} \sum_{j \in L} (2 \sum_{k \in L} u_{ijk} A_{jk} - \alpha s_j + v_{ij}) x_{ij},$$

subject to (4), (6).

Sub-problem 4:

$$Z_{sub4} = \min \left\{ - \sum_{i \in F} \sum_{j \in L} \sum_{k \in L} u_{ijk} A_{jk} \right\},$$

According to the weak Lagrangean duality theorem, we know that $Z_D(s, t, u, v)$ is a lower bound on Z_{IP} , for any $(s, t, u, v) \geq 0$. Thus, the dual problem Z_D is used to calculate the tightest lower bound

$$Z_D = \max_{s, t, u, v \geq 0} Z_D(s, t, u, v)$$

The sub-gradient method is used to solve the Z_D problem.

By solving the dual problem, we can derive a set of decision variables that may be appropriate for the primal heuristic algorithm, the output of which provides the upper bound of the primal problem Z_{IP} . Our heuristic algorithm is detailed in Figure 1.

```

Step1 Allocate
s ← 1;
for j = 1 to n {
  foreach (i ∈ F, xij = 1) { copy xij to tij; }
  foreach (i ∈ F, tij = 1) { xsj ← tij; s++; }
}
Step2 Adjust
for j = 1 to n {
  foreach (i ∈ F, xij = 1) {
    find ℓ, that satisfies ℓ ∈ {1, i-1} and xℓj = 0
    if xℓj ∩ xℓk = ∅, ∀k < j, then
      move xij to xℓj;
      set yi ← 1;
    endif
  }
}
compute CURRENT ← sumover(yi), ∀i ∈ F;
Step3 Reassign
for j = 1 to n {
  foreach (i ∈ F, xij = 1) {
    find ℓ, that satisfies ℓ ∈ {1, i-1} and xℓj = 0;
    if xℓj ∩ xℓk ≠ ∅, ∀k < j, then
      set xij ← 1;
      SATISFIED ← VERIFY_CONSTRAINT(xij);
      if SATISFIED = false, then
        restore xij, Bij to their original values;
      else
        // Will it violate the constraint to stay at
        // the current position?
        SATISFIED ← VERIFY_CONSTRAINT(xij);
        if SATISFIED = false, then
          find ℓ, that satisfies ℓ ∈ {i-1, n} and xℓj = 0;
          move xij to xℓj;
        endif
      endif
    endif
  }
}

```

```

    endif
  }
}
Step4 check stop condition
compute NEW ← sum over yi, ∀ i ∈ F;
if NEW < CURRENT, then go to step3 else stop;

```

Figure 1 The Getting Primal Heuristic Algorithm

The heuristic algorithm consists of 4 steps. Step1 staggers every reader's slots by reassigning decision variables x_{ij} in a straightforward assignment which produces an upper bound of cycle ($\sum y_i$). According to the nature of the Lagrangean approach, we know that Constraint (4) is satisfied during the computation. Thus, we know the value of $\sum y_i$ will be equal to $\sum c_j$. Step2 adjusts the array of x_{ij} , by satisfying $x_{ij} \cap x_{ik} = \emptyset$, such that $\sum y_i$ is the minimum and there is no overlapping (i.e., all B_{ij} are equal to 0). Every reader compares itself with its left-hand side (LHS) readers, which reduces the amount of computation Step3 moves the assigned slot of every reader forward from the current position without violating Constraints (1) and (2), by calling the subroutine VERIFY_CONSTRAINT. If the slot is unable to move ahead, then it is considered to be moving backward because its current position might violate the constraints due to previous adjustments of its LHS reader. Step4 is the stop condition, as no further improvement is possible.

Our experiment shows that the Lagrangean approach is ineffective, since the solution gaps are too high. In the development of the LR method, we found that our heuristic algorithm is efficient in getting the primal upper bound of y_i , x_{ij} and B_{ij} . We compared the result of our heuristic algorithm and those from OPL CPLEX and found they were identical.

4. Computation Experiments

Figure 2 shows three irregular reader networks, taken from [1], which we used for the computation experiments. Two slot requirement groups of c_j , low (request ≤ 3) and high (request ≥ 8), were compiled for the experiment, as shown in Table A. We tested the networks with the threshold values of α and β in (0.5, 0.1), (0.5, 1) and (0, 0) respectively. (0, 0) indicates that the problem is exactly an n-coloring problem. We modified the heuristic algorithm as an independent program, which computes each result only in seconds. The experiment results are shown in Table B.

We used OPL CPLEX to run two small size irregular networks, NW1 and NW2, with low slot requirements and with three distinct (α, β) value pairs for comparison. The computational results of optimal solution y_i , derived by CPLEX, are identical to those of the heuristic algorithm.

5. Conclusion and Future Work

In this paper, we propose a mathematical model for reader networks in which the readers overlap, and derive a near optimal heuristic algorithm based on the solution procedure of the Lagrangean method. Our algorithm, which is modified so that it can be run as an independent slot assignment procedure in an RFID reader network, can effectively calculate the values of three decision variables: y , B , and x . The algorithm performs well in small networks but the computation time increases exponentially because the problem is NP-completed. The Lagrangean approach is proved to be ineffective, because the solution gaps in the optimal value vary between 40% and 76% in our experiment (as shown in Table B). It could be that the nature of the problem's nature makes the Lagrangean approach unsuitable. In our future work, we will consider a simulated annealing approach to find a better solution.

6. References

- [1] Frank Yeong-Shun Lin, "Quasi-static Channel Assignment Algorithms for Wireless Communication Network," Information Networking, 1998. (ICOIN-12) Proceedings., Twelfth International Conference on 21-23 Jan. 1998 pp. 434 – 437
- [2] Engels, D.W., Sarma, S.E., "The Reader Collision Problem," Systems, Man and Cybernetics, 2002 IEEE International Conference on Volume 3, pp. 6-9, 2002
- [3] Waldrop, J.; Engels, D.W.; Sarma, S.E., "Colorwave: a MAC for RFID reader networks," Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE Volume 3, pp.1701 – 1704, 16-20 March 2003
- [4] Sun Microsystems, "Sun EPC Network Architecture, A technical White paper," February 2004, http://www.sun.com/aboutsun/media/presskits/networkcomputing04q2/epc_whitepaper.pdf
- [5] A. M. Geoffrion, "Lagrangean relaxation and its use in integer programming," Math Programming Study, Vol 2, pp.82-114, 1974
- [6] M. L. Fisher, "The Lagrangean relaxation method for solving integer programming problems," Management Science, vol.27, pp 1-18, 1981.
- [7] Philipose, M. Smith, J.R.Jiang, B. Mamishev, A. Sumit Roy Sundara-Rajan, K., "Battery-free wireless identification and sensing," Pervasive Computing, IEEE, Jan.-March 2005, Volume: 4, Issue: 1 pp. 37- 45
- [8] W.K.Hale, "Frequency Assignment: Theory and Applications," Proc, IEEE, vol.68, pp. 1497-1514, 1980.

Table A, The Requirement Matrix Of Low And High

	c_j :low requirement	c_j :high requirement
NW1	2,3,2,1,2,3,1,3,2,3	8,15,14,18,14,16,15,13,14,13
NW2	2,3,1,3,2,3,2,1,1,2,1,3	9,11,11,13,22,23,22,7,21,32,11,13
NW3	2,3,1,3,2,3,2,1,1,2,1,3,2,3,1,3,2,3,2,2,2	14,13,16,13,15,16,12,14,23,13,17,23,14,13,14,8,15,13,12,12,14

Table B, The Upper Bound Values Of y_i and The Solution Gaps of Lagrangean Approach,

			UB of y	LR result	Gap
NW1	$\alpha=0.5$ $\beta=0.1$	Low	7	2.723965	57%
		High	39	14.88	61%
	$\alpha=0.5$ $\beta=1$	Low	7	2.53	57%
		High	37	14.3996	60%
	$\alpha=0$ $\beta=0$	Low	10	2.53	57%
		High	59	14.3996	60%
NW2	$\alpha=0.5$ $\beta=0.1$	Low	6	2.57	50%
		High	55	21.4579	60%
	$\alpha=0.5$ $\beta=1$	Low	5	2.3348	40%
		High	55	19.5	64%
	$\alpha=0$ $\beta=0$	Low	6	2.3348	40%
		High	76	19.5	64%
NW3	$\alpha=0.5$ $\beta=0.1$	Low	7	2.179	57%
		High	50	15.22514	68%
	$\alpha=0.5$ $\beta=1$	Low	7	2.0095	57%
		High	45	10.799	76%
	$\alpha=0$ $\beta=0$	Low	10	2.0095	57%
		High	61	10.799	76%

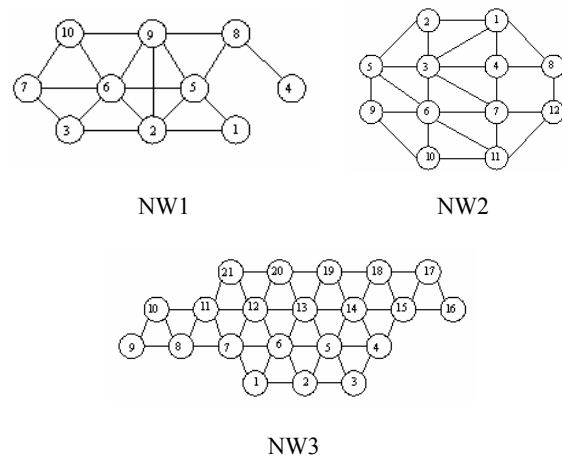


Figure 2, 3 Irregular Networks