Scheduling in Wireless Sensor Networks WSNs

Bc. Tomáš Novotný
Thursday 16:15
Open Informatic
novott16@fel.cvut.cz

Abstract

***Abstract* - The paper contains definition, description of implementation and evaluation of superframe multichannel communication of low-power wireless sensor network for heuristic algorithm and exact algorithm.**

I. Assignment

A. Problem Statement

In modern houses we are surrounded by sensors, which transmits the readings by radio to central computer. These devices should be as small as possible, so various thing of daily use could contain them. And every circuit need power to operate and communicate, so for small device we could fit only small battery and most energy is used for the communication. We can reduce battery consumption by arranging sensors to the network, so the sensor can pass information to the nearest sensor and don't have to waste energy to reach the destination point. By arranging sensors into network we create directed acyclic graph, each sensor transmit information for fix period of time, also there is matrix defining which devices cannot transmit on same channel at the same time to prevent collisions. So the algorithm input is directed acyclic graph *G(V,E)* consisted of *n* nodes, where *V* is set of sensors and *E* is presence relation between devices (i.e. when edge leaves V_i and enters V_j, then V_i end it's transmission before vj start to transmit), Each sensor can transmit only once per period and it transmits last *SD* (superframe duration), matrix C with $|V|$ * $|V|$ dimension, where C_{ij} = *1* means device V_i and V_j are not allowed to transmit on same channel at same time, C_{ij} = *0* means that devices are allowed to use simultaneously same channel without collision. And *H* channels in which can be transmitted. The task for algorithm is to find schedule such that the schedule period length is minimized. (i.e. the time from start of first transmission in network to the end of last transmission in network is minimal)

1) Example problem: We are given graph *G*, matrix *c*, number of channels *H* and vecort *SD*

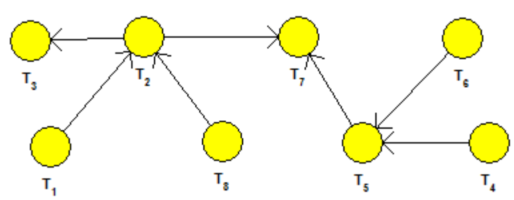

Figure 1. Nodes in network

$$C = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Figure 2. Matrix C defining transmission relation

$$H = 2$$
$$SD = \begin{bmatrix} 4 & 3 & 8 & 2 & 5 & 5 & 7 & 1 \end{bmatrix}$$

Figure 3. Vector SD defining transmission duration and number of channels

B. Problem Categorization

The general problem is Parallel scheduling problem combined with Constraint satisfaction problem. Parallel scheduling problem scheduling the transmissins on channels. This problem is similar as TSP, when tasks are cities and processor is salesman. Since TSP is proven to be NP-hard probelm and in Parallel scheduling problem we have multiple processors e.g. multiple sales man, this problem is also NP-hard. Constraint satisfaction problem is represented by transmission parallelism constrain.

II. Related works

Problem of superframe multichannel wireless sensor network scheduling is shared with article [1]. But algorithm in article [1] is designed as online and is reflecting wireless standards and collisions on real senosr network (IEEE 802.15.4/ZigBee sensor network). This paper is more theoretical and algorithm is designed as offline and our network don't have characteristic of cluster-tree. Article [2] descibes algorithm which tries to maximaze the life-span of devices on tree-clueste ZigBee network. In this paper we tries to minimaze the overall time for tramsmitting the information. Another work [3] , which describes superframe scheduling usning two channels on HART wireless protocol. Also fixed ration od packet-loss is considered.

III. Problem solution

A. Design

1) ILP: Let's sumarize the problem in ILP. We are trying to schedule the transmissions as early as possible. So we have to minimalize time of end of each task, which is start *S* plus duraiton *SD* (1).

$$minimize \sum_{i=1}^{n} S_i + SD_i \tag{1}$$

In order to schedule transmission, we have to give it channel *H* and start time *S*, so that 4 conditions are not met together.

1) Transmission *i* is can be transmitted at same time as transmission *j*. This condition si defined by matrix *C*.
2) Transmission *i* and transmission *j* are not on same channel.
3) Transmission *i* and transmission *j* ane not overlapping in time.
4) Transmission is not scheduled before all predecesting transmissions have ended.

This ensure following constrain (2), where *C* is parameter given at begining, *Kc* is matrix of binary variables for each two transmissions, which says if transmission *i* is on same channel as transmission *j*, and *Tc* is similar matrix, but for overlapping in time.

$$C_{i,j} + Kc_{i,j} + Tc_{i,j} => 2 \tag{2}$$

Decision of variable *Kc* is based on two constrains, because ILP have to be declared in language of inequations. So constarins (3) and (4) decides if channel is the same and Big M method and binary variable $Kb_{i,j}$ is used to make OR effect. If channels are the same both constrains could not be satisfied and then vairable $Kc_{i,j}$ is setted to *true*;

$$((1 - Kb_{i,j} + Kc_{i,j}) * -M) + H_i - H_j <= -1 \tag{3}$$

$$((Kb_{i,j} + Kc_{i,j}) * -M) + H_j - H_i <= -1 \tag{4}$$

Same principle as for constains (3), (4) is used for constrains 5) and (6). But we are checking the end time of transmission *i* is less or equal to start of transmission *j*.

$$((1 - Tb_{i,j} + Tc_{i,j}) * M) + S_j + SD_j - S_i <= 0 \tag{5}$$

$$((Tb_{i,j} + Tc_{i,j}) * M) + S_i + SD_i - S_j <= 0 \tag{6}$$

Last condition is satisfied by constrain (7). *E* is adjacency matrix of graph. If there is no edge between *i* and *j* the Big M satisfies the constrain, otherwise end of transmission *i* must be before start of transmission *j*.

$$((1 - E_{i,j}) * M) + S_i + SD_i <= S_j \tag{7}$$

2) Heuristic algorithm: Let's describe heurictic algorithm which schedule the transmissions step by step. As input we have matrix C, graph G represented by adjacency matrix and vector SD.

1) First we select transmission with no predecessors, the leaves of the graph, and put them in the set of schedulable transmission.
2) Start iterating thrue the set of schedulable transmission and try to schedule the fist transmission.
3) To determinate channel and realease time we have to check every channel and already scheduled transmissions in the channels. So we find all the cheduled transmissions which have collision with transmission we are trying to schedule. That means $C_{i,j} = 1$. We can schedule our transmission after the latest transmission in each channel. So we find already scheduled transmission with collision and also the latest from each channel. And schedule our transmission to channel with the earliest ending transmission of these latest transmissions.
4) Every time we schedule transmission, we remove it from the graph and add new leaves to schedulable set.
5) Continue scheduling until all tasks are scheduled.

This algorithm is terminate, because in every iteration we schedule one task.

B. Implementation

// For both implemetations I kept input data of examples in two matrices and vector. This allowed me to share exaple probles with minimal effort of reformating.

1) ILP: For solving the problem as exact ILP problem I used Gusek, which is IDE for solving MILP and ILP problems based on GLPK solver. As interesting part of my implementation of given problem I would mark the usage of multiple binary variables in Big M method. This way i was able to linked multiple constrains together. Algorithm prints reasul as vector of values separated by tabs for each transmission in format: id, start, duration, end, channel.

2) Heuristic algorithm: For implementing heuristic algorithm I chose Java SE (version 1.8). As I'm familiar with this language and I appreciate work of garbage collector. I used standart java ArrayList collections to store class Task, which represented transmission. Class Task contained all variables about the transmission, that's mean id, channel, start, duration, number of predecessors and list of descendants, which were notified about removing they predecessor from graph and optionali returned and addet to collection of waiting task, if they became leaves. Task ready for scheduling were stored in HasSet. I chose HasHSet for its's speed, when removing Task from this set. Algorithm prints reasul as vector of values separated by tabs for each transmission in format: id, start, duration, end, channel.

IV. Experiments

A. Benchmark Settings

I tested my algoritms on desktop PC. For testing I used several diffently large instances of the problem and analyzed the load on HW.

- CPU: Intel i5 3,3 GHz(4cores, 4threads)
- RAM: 16 GB
- Windows 10 (64-bit)

1) ILP: Gusek IDE it self mesures memory used by the GLPK and running time of instance. Also I checked system performance monitor.

2) Heuristic algorithm: I use IdeaIDE for Java development and Memory analyzer from Ecpilse fondation, for analyzing Java heap. Also I checked system performance monitor.

B. Results

// Both algorithm gave same minimal duration of the whole tansmission window in cases that exact algorithm fihished calculation. But scheduling of praticulars transmissions was different. Heuristic algorithm was inserting task in random order. It's result of iterating thrue HasHSet, which was uset to store the Task, when the task waited for shecduling. But heuristic algorithm in Java proven to be faster and less memory demanding. When there is *n/a* in table, that means calculation wasn't finished even after 15 minutes and instance was killed. On the other hand heurisit algorithm fisnished always within few seconds. That's because each time algorithm checked all the transmission 1 transmission was scheduled. But for exact algorithm tree of variants each iteration grows and grows. Heuristic algorithm memory complexity was linear depending on number of created object, on the orther hand branch & bound in excast algorithm growed much faster.

Obsah...

Type	Nodes	Channels	Time	Memory
Heuristic	7	2	<0s	2 MB
Exact	7	2	<0s	<1 MB
Heuristic	10	5	<0s	4 MB
Exact	10	5	17 s	5 MB
Heuristic	15	3	<0s	8 MB
Exact	15	3	n/a	100 MB
Heuristic	20	5	<0s	12 MB
Exact	20	5	n/a	100 MB
Heuristic	100	15	1s	53 MB
Exact	100	15	n/a	100 MB

Table I
TABLE OF RESULTS OF FOR TESTING INSTANCES

C. Discussion

I expected that heuristic algorithm would be faster than exact algorithm, but I was surprised by the number of nodes and channels for which exact algorithm fails to give solution in relatively great amount of time. I understand that the problem is Np-hard, but didn't expect that branch & bound method is that slow with comparasion with algorithm, which may return sub-optimal solution for greater problems.

V. CONCLUSION

I was able to create and compare two algorithm with different aproach to the problem. Each functioning flawlessly on small tasks. But for larger task the exact algorithm proven itself to be unusable, due computation requirements. And so the algorithm which may returns sub-optimal solution is usable even for larger tasks, without any overwhelming performance requirements. I was not able to verify if result returned by heuristic algorithm for larger tasks are optimal, but we can consider them as sub-optimal.

REFERENCES

[1] Toscano, E.; Lo Bello, L., "Multichannel Superframe Scheduling for IEEE 802.15.4 Industrial Wireless Sensor Networks," in *Industrial Informatics*, IEEE Transactions on ,vol.8, no.2, pp.337-350, May 2012

[2] A. Ahmad, Z. Hanzalek and C. Hanen, "A polynomial scheduling algorithm for IEEE 802.15.4/ ZigBee cluster tree WSN with one collision domain and period crossing constraint," *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, Barcelona, 2014, pp. 1-8. doi: 10.1109/ETFA.2014.7005182

[3] G. Fiore, V. Ercoli, A. J. Isaksson, K. Landernäs and M. D. Di Benedetto, "Multihop multi-channel scheduling for wireless control in WirelessHART networks," *Emerging Technologies & Factory Automation*, 2009. ETFA 2009. IEEE Conference on, Mallorca, 2009, pp. 1-8. doi: 10.1109/ETFA.2009.5347073