Xu *et al. EURASIP Journal on Wireless Communications and Networking* 2014, **2014**:214
http://jwcn.eurasipjournals.com/content/2014/1/214

EURASIP Journal on
Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

CGA-based deadlock solving strategies towards vehicle sensing systems

Jianlong Xu, Zibin Zheng* and Michael R Lyu

Abstract

Vehicle sensing system is an important research topic in the research field of Internet-of-Vehicles (IoV). Reliability and real-time performance of vehicle sensing systems are greatly influenced when deadlock happens. When a deadlock is detected, identifying the optimal deadlock solving strategy can ensure that the system goes back to normal state quickly. In order to address this issue, this paper proposes an efficient deadlock solving method. Firstly, the deadlock problem in a vehicle sensing system is analyzed based on four deadlock occurring conditions (i.e., mutual exclusion, hold and wait, no preemption, and circular wait). Secondly, an optimization model is built to combine the quantity and cost of tasks in vehicle sensing systems. After that, a co-evolutionary genetic algorithm (CGA) is developed to search the optimal deadlock solving strategy. Finally, experiments by simulation are conducted and the experimental results show the efficiency of the proposed deadlock solving method for vehicle sensing systems.

Keywords: Vehicle sensing system; Deadlock solving; Co-evolutionary genetic algorithm

1 Introduction

With the development of intelligent transport systems (ITSs), Internet-of-Vehicles (IoV) techniques are booming rapidly in recent years. IoV enables information sharing and gathering on vehicles, roads, and their surrounds. Building highly efficient and intelligent vehicles is very important for automobile industry [1,2]. Vehicle sensing systems, which are important research topics in IoV, have attracted high attentions of many research institutions worldwide [3,4]. Vehicles are equipped with hundreds of intelligent sensing devices, such as GPS-based sensors, video cameras, and other physical and chemical sensors, which are distributed over their chassis, powertrain, body, wheel areas, and so on [5]. Some sensors are independent, while others communicate with each other through wired or wireless networks, such as controller area network (CAN), Bluetooth, ZigBee, LowPan, etc. These sensing devices form a complex heterogeneous network inside the vehicle. They are capable of collecting, processing, storing, and transferring information from one node to another, and they perform a lot of tasks, including monitoring temperature, pressure, humidity, vehicular motion state, noise levels, and so on. Generally, new and pertinent information will be obtained by collecting and processing the raw sensor data. For example, three parameters including motion attitude, dynamic load, and braking performance can be obtained by utilizing data from acceleration sensors, temperature sensors, pressure sensors, and GPS-based sensors, which are embedded in the vehicle wheel, body, and suspension mechanism [6,7]. In these cases, the collecting and processing tasks which request resources (e.g. communication resource, storage resource, etc.) must wait until they acquire all the requested resources. If the requested resources are held by other tasks, deadlocks may happen in the system, which will greatly influence the system reliability and real-time performance. Therefore, the deadlock problems must be taken into consideration in the design and development phases of vehicle sensing systems.

In vehicle sensing systems, because of the variety of tasks and resources, the deadlock is hard to be predicted. The efficiency of existing deadlock solving methods needs to be improved. Different from traditional software systems, vehicle sensing systems are safety critical, where deadlock problems may lead to serious and fatal consequences. Moreover, vehicle sensing systems have a

*Correspondence: zbzheng@cse.cuhk.edu.hk
Shenzhen Key Laboratory of Rich Media Big Data Analytics and Applications, Shenzhen Research Institute, The Chinese University of Hong Kong, Beijing, China

number of sensors, which are shared among different tasks, making deadlock problems more notable. Existing work [8] has mentioned the idea of enhancing the quality of hardware to improve the safety and reliability of vehicles. However, this hardware-based approach is expensive and not practical. Therefore, in order to find better ways to make sure that the system is more stable and reliable, more practical solutions are urgently needed for vehicle sensing systems.

In order to address this critical challenge, this paper proposes an efficient deadlock solving method, which aims at improving the efficiency of deadlock solving mechanisms in vehicle sensing systems. In summary, the major contributions of this paper include: 1) we analyze the deadlock problem in vehicle sensing systems based on the deadlock occurring conditions; and 2) we build a deadlock solving optimization model and propose a deadlock solving method based on co-evolutionary genetic algorithm for vehicle sensing systems, which can quickly solve the deadlock problem and ensure the minimum cost.

The rest of this paper is organized as follows: Section 2 summarizes related work on vehicle sensing systems and deadlock solving method. Section 3 explains the deadlock analysis model in the vehicle sensing systems. Section 4 first describes the deadlock solving optimization problem and then proposes an optimization method based on co-evolutionary genetic algorithm to search the optimal deadlock solving strategy. Afterwards, Section 5 demonstrates the experimental results. Finally, Section 6 concludes the paper and discusses the future work.

2 Related work

2.1 Vehicle sensing systems

Vehicle sensing systems have been one of the most active research areas in the automotive field over the past and current decades [5]. In terms of safety, pollutants, greenhouse gases emissions, comfort and energy consumption, etc., research in vehicle sensing systems is conducted from various aspects, aiming to improve the system reliability. Yimetal [8] proposed a Smart Car Sensor Network (SCSN) to improve the communication reliability for sensor networks. Taghvaeeyan et al. [9] proposed position estimation algorithms based on the use of anisotropic magnetoresistive sensors for imminent crash detection in cars. In the work of [10], Taghvaeeyan further proposed a two-dimensional sensor system for automotive crash prediction. Georgakosetal [11] also discussed the reliability challenges for electric vehicles from devices to architecture and systems software. Xiuying et al. [12] focused on a velocity monitoring system based on the multi-sensor data fusion method, which was applied in car navigation and positioning systems. With the development of vehicle sensing technology, reliability of the vehicle sensing system is becoming more and more important.

2.2 Deadlock solving method

Deadlock is one of the most important problems in both centralized and distributed systems. A deadlock is a situation in which two or more competing actions are waiting for each other to finish, and thus neither ever does [13,14].

Deadlock problem solving strategies can be classified into active strategies and passive strategies. Active strategies solve a deadlock before it occurs by online or offline methods. Active strategies include deadlock prevention strategy and deadlock avoiding strategy. The main purpose of deadlock prevention strategy is to ensure that the system always stays away from a deadlock state. It uses offline calculation mechanism to control the request and allocation of resource through imposing restrictions to the system. Various deadlock prevention strategies are proposed in recent years. Yi-Sheng et al. [15] proposed a deadlock prevention algorithm for sequence resource allocation systems, employing Petri nets to build and describe the systems of simple sequential the processes with resources (S^3PGR) model. On the other hand, for the deadlock avoiding strategy, system state is monitored continuously in the process of system operation. The system scheduling policy will judge whether a process will lead to a deadlock or not and then decides the next system operation process. Ballaletal [16] proposed the MAXWIP (Max a work in progress) algorithm for deadlock avoidance in mobile wireless sensor network monitoring systems, which was described as Free Choice Multi-Reentrant Flow Line (FMRF) systems.The advantage of this method is that system operating efficiency can be kept in the greatest degree. But it must obtain all the system reachable states first, which is not realistic in medium- or large-scale systems.

Passive strategies solve the deadlock problem when a deadlock occurs, including deadlock detection strategy and deadlock relieving strategy. The efficiency of passive strategies depends on the speed of deadlock detection and deadlock relieving. After a deadlock is detected, the system can be unlocked by automatic or manual methods. While this approach tends to achieve high real-time performance and resource utilization, it requires full understanding of the deadlock. Abd El-Gwad et al. [17] proposed a deadlock detection protocol based on threads, which schedules the threads in order to detect which thread would initiate the deadlock. Aydin Aybar et al. [18] presented an approach to design a supervisory controller for a timed Petri nets (TPNs) to avoid deadlock by using the method of stretching. The presented approach determines the least restrictive controller which guarantees deadlock avoidance, whenever such a controller exists. Steghofer et al. [19] described a distributed deadlock avoidance algorithm for self-organizing resource flow systems. The algorithm leverages implicit local knowledge about the system structure and uses a simple coordination mechanism to detect loops in the resource flow. But this algorithm must use specific system architecture knowledge and its generality is not strong. Xing et al. [20] proposed a deadlock-free genetic scheduling algorithm to optimize the performance of automated manufacturing systems based on deadlock control policy, which embedded the optimal deadlock avoidance policy into the genetic algorithm and used the one-step look-ahead method in the optimal deadlock control policy.

In the above research investigations, researchers focus on vehicle sensing systems or deadlock solving methods respectively. There is limited study on the deadlock problems in vehicle sensing systems. In order to improve the system reliability, we propose a deadlock solving method for vehicle sensing systems in this paper.

3 Vehicle sensing system deadlock analysis

Figure 1 shows the schematic of the vehicle sensing system. There are many sensing resources (e.g. vibration sensors, gyroscopic sensors, ultrasonic sensors, etc.) located in different parts of the vehicle for different functions.

We assume that the vehicle sensing system tasks include information collection, transmission, computation, converging, and control, etc. The system resources include sensing devices, communication resources, storage devices, information transmission equipments, computing devices, actuators, and so on. Figure 2 shows the system task flow and related resources.

In Figure 2, the system task set and resources set are expressed as $T = \{T_1, T_2, \ldots, T_n\}$ and $R = \{R_1, R_2, \ldots, R_m\}$, respectively. Let $R_1, R_2, \ldots$ and R_m represent different kinds of resources. Each kind of resources may contain more than one device. The resource allocation relationship can be expressed as a graph G:

$$G =< V, E > \tag{1}$$

In Equation 1, V includes T and R and set E is constituted by all the ordered pair (T_i, R_j) and (R_j, T_i).

More specifically, $V = \{T_1, T_2, \ldots, T_n, R_1, R_2, \ldots, R_m\}$, $E = \left\{\left(T_i, R_j\right), \left(R_i, T_j\right) | T_i \in T, R_j \in R\right\}$. If $\left\{\left(T_i, R_j\right)\right\} \in E$, it expresses that there is a directed edge from T_i to R_j, which means task T_i requests resource R_j. If $\left\{\left(R_j, T_i\right)\right\} \in E$, then the directed edge is from R_j to R_i, which means R_j is held by T_i.

Previous research indicates that four conditions may lead to a deadlock, which are *mutual exclusion, hold and wait, no preemption, and circular wait* [13,21]. In vehicle sensing systems, corresponding deadlock conditions are described as follows.

Mutual exclusion is expressed as $\left(\exists \left(R_j, T_i\right) \in E\right) \wedge \left(\nexists \left(R_j, T_l\right) \in E\right), l \neq i$, which means that system resource R_j can only be held by a task (e.g., sensor information collection task, etc.) or keep idle; other tasks cannot hold this resource at the same time.

Hold and wait can be expressed as $\left(\left(R_j, T_i\right) \in E\right) \wedge ((T_i, R_l) \in E) \wedge ((R_l, T_l) \in E), l \neq i$,which means T_i has held at least one resource R_j and requested a new resource R_l, but R_l has been held by T_l. The request will be blocked because the resource R_l has not been released.

No preemption is expressed as $(t < t_i) \wedge (\exists (R_j, T_l) \in E)$. t_i is the time of T_i holds R_j. It means sensing resources R_j cannot be held or deprived by other tasks before task T_i releases it.

Circular wait can be expressed as $\exists \{T = \{T_1, T_2, \ldots, T_n\}, R = \{R_1, R_2, \ldots, R_m\}\}, \exists ((R_1, T_1) \in E)) \wedge ((R_2, T_2) \in E) \wedge \ldots ((R_n, R_n) \in E)$ and $\exists ((T_1, R_2) \in E)) \wedge ((T_2, R_3) \in$

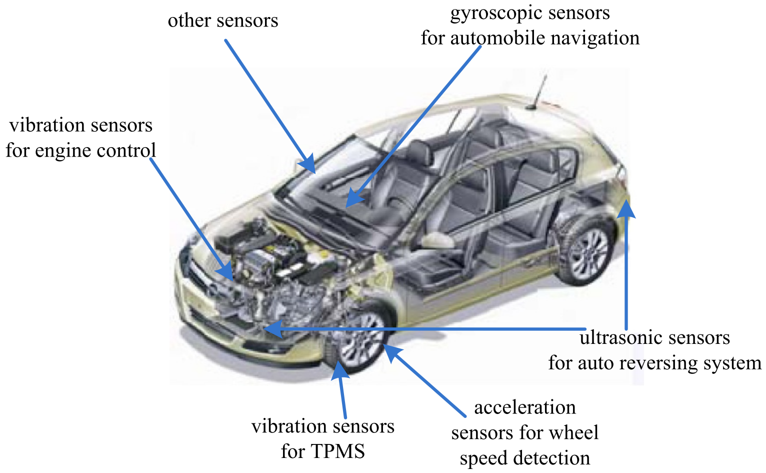

Figure 1 The schematic of vehicle sensing system.

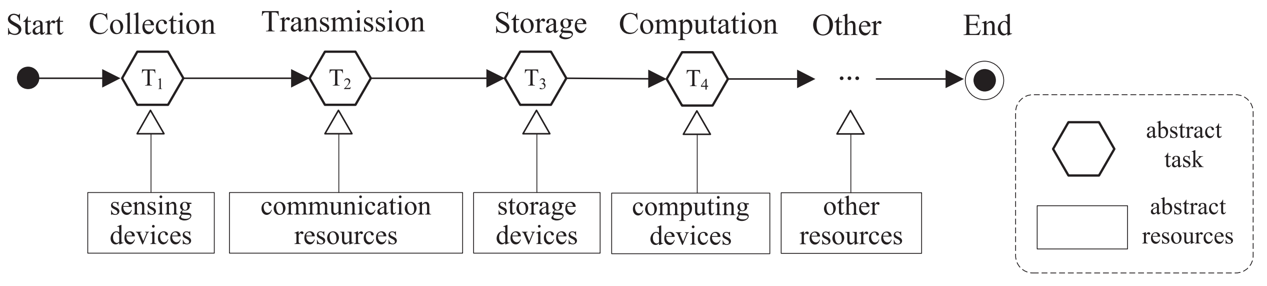

Figure 2 System tasks flow and related resources.

$E) \wedge \ldots ((T_n, R_1) \in E)$. It means that T_1 is waiting for R_2 which is held by T_2, T_2 is waiting for R_3 which is held by $T_3, \ldots$, and T_n is waiting for R_1 which is held by T_1. Thus, it results in a circular chain.

4 Deadlock solving optimization method based on co-evolutionary genetic algorithm

When a deadlock is detected, a deadlock solving mechanism is needed to ensure that the system goes back to normal. Generally, the method of solving deadlock is to abort one or more tasks and consequently release the held resources, which can break the state of circular wait. This section first describes the deadlock solving optimization problem and then proposes an optimization method based on co-evolutionary genetic algorithm.

4.1 Deadlock solving optimization problem of vehicle sensing systems

Figure 3 shows n tasks and m resources falling into deadlock in a vehicle sensing system. The tasks and resources are in the state of circular wait. In the figure, the black arrows mean resources are held by tasks while white arrows mean tasks request resources. We assume that the quantity of held or requested resource R_j is N_j, which includes held resources (HRs), requested resources (RRs), and the resources being held and requested at the same time (HR&RR). The quantity of the ith HR, RR, and HR&RR are denoted as $N_{\mathrm{HR}i}$, $N_{\mathrm{RR}i}$, and N_{si}, respectively.

If HR&RR exists, then

$$N_{\mathrm{HR}} + N_{\mathrm{RR}} > N_j \tag{2}$$

In order to break the deadlock, it is necessary to release HR&RR held by some tasks. When deadlock happens and the location is unknown, N_j resources are involved, where some of them need to be reallocated. It is complex to decide which resources should be released from the tasks. So, the deadlock solving strategy is not unique.

In Figure 4, R_1 and R_2 are different resources. R_1 and R_2 both have four sensing devices. The quantity of sensing devices in R_1 held by T_1, T_2 is $1, 1$, respectively. The quantity of sensing devices in R_1 requested by T_3 is 2. While the quantity of R_2 held by T_3 is 2, the quantity requested by T_1 and T_2 is 1 and 1, respectively. Because the sensing devices in R_1 and R_2 are both being held and requested, T_1, T_2, and T_3 may fall into the state of circular wait.

Since there are three tasks, there can be seven possible deadlock solving strategies, which are shown in Table 1.

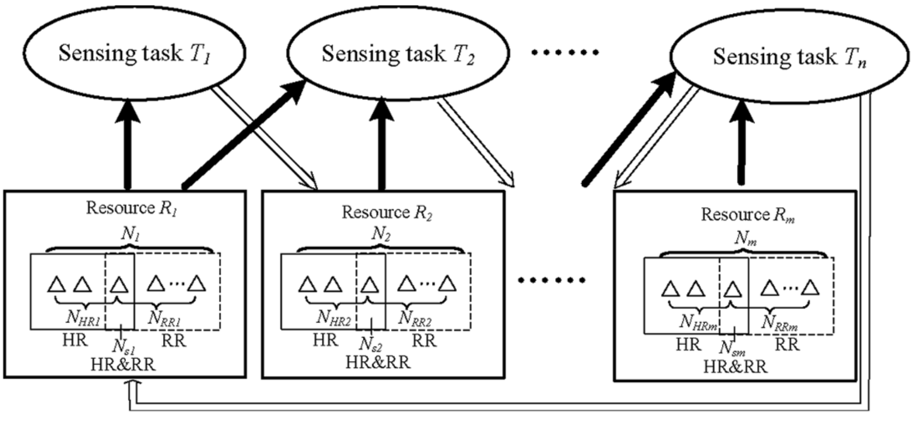

Figure 3 n tasks and m resources fall into deadlock.

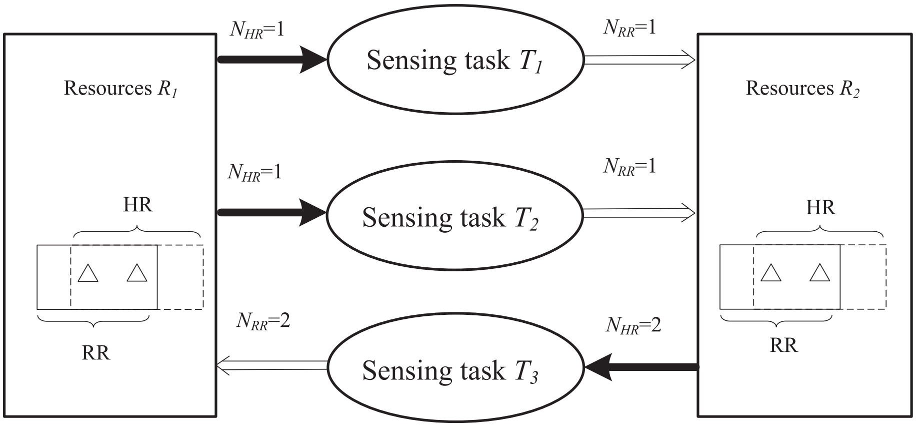

Figure 4 Three tasks and two resources fall into deadlock.

Obviously, the deadlock solving strategy is not unique, and the optimal strategy is the one that aborts the least number of tasks. However, other factors (e.g., cost, operating state, urgency degree, priority, etc.) may be different with each task. For example, if the costs of aborting T_1, T_2, and T_3 are 2, 3, and 4, respectively, then the cost of aborting T_1 and T_2 is 5, but the cost of aborting T_3 is 4, so the latter is a better strategy. In this paper, we try to identify the optimal deadline solving strategy considering both quantity of aborted tasks and their cost.

Due to the complexity and variability of relationship between tasks and resources in the vehicle sensing system, the resources are held or requested by multitasking, which may lead to deadlocks. To solve the deadlocks and at the same time to minimize the quantity of aborted tasks and the cost, the objective function of the deadlock solving optimization model can be described as:

$$\begin{aligned} &\min \sum_{i=1}^{n} \mathrm{cost}_i \times S_i \\ &\text{s.t.} \sum N_{T_i \leftarrow R_j} + \sum N_{T_i \rightarrow R_j} \leqslant N_j \end{aligned} \tag{3}$$

Table 1 Seven strategies for deadlock solving

Strategies	Can solve the deadlock or not?
Abort T_1	No
Abort T_2	No
Abort T_3	Yes
Abort T_1 and T_2	Yes
Abort T_2 and T_3	Yes
Abort T_1 and T_3	Yes
Abort T_1, T_2, and T_3	Yes

In Equation 3: $N_{T_i \leftarrow R_j}$ denotes the number of resource R_j held by task T_i, $N_{T_i \rightarrow R_j}$ denotes the number of resource R_j requested by task T_i, N_j is the number of the same kinds of factors in R_j, cost_i is the cost of aborting the task T_i, S_i is the state of T_i, $S_i \in \{0, 1\}$. We assume 0 stands for aborting a task while 1 stands for not aborting a task. For this model, there are $2n - 1$ kinds of possible strategies which need to be checked. Some strategies can solve the deadlock problem while others cannot. Since the optimal solution search space is very large, it is time consuming to obtain the optimal solution using an exhaustive method. To address this problem, we propose a deadlock solving method based on a genetic algorithm to enhance the efficiency.

4.2 Deadlock solving optimization method based on co-evolutionary genetic algorithm

Genetic algorithm can search through simulating natural selection and evolution, and they enjoy excellent global optimal convergence ability [22]. It uses basic operators, namely crossover and mutation, and is based on the evolution of a population of similar individuals (species from the same class) [23]. Each individual can be a solution for the target problem we want to solve. Basic genetic algorithm has been a good way of solving many optimization problems and got comparative desirable results. However, it also has some shortcomings in some cases which are intrinsically distributed. In other words, it can be seen as several independent entities with their own goals interacting with each other, which may go against the whole goal of solution. To address these kinds of problems, in this paper, we employ co-evolutionary genetic algorithm to solve the deadlock strategies choices. In our deadlock solving optimization method, each deadlock solving strategies are considered as individuals. The population, namely, a collection of such individuals, will encounter genetic operators such as crossover and

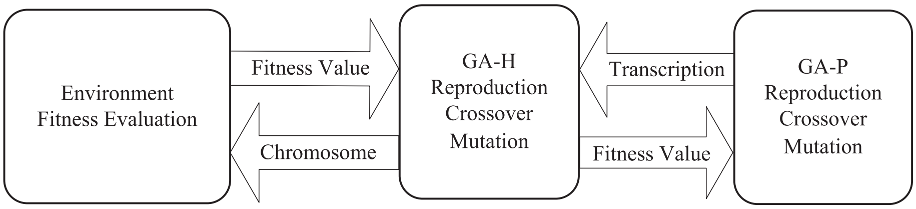

Figure 5 The frame of co-evolutionary genetic algorithm.

mutation in each generation. After a number of iterations, the best individual obtained is the best deadlock strategy.

4.2.1 Outline of the co-evolutionary genetic algorithm

The co-evolutionary genetic algorithm extends the basic genetic algorithm (GA) and makes better use of effective genetic information given by a direct search for useful schemata. That may make the search ability of GA more efficient and it has good speed in the application of searching and optimization [24,25]. The method consists of two GA models interacting with each other. The first GA model is named host genetic algorithm (GA-H) and the second GA model is named parasite genetic algorithm (GA-P). As shown in Figure 5, GA-H is the traditional GA model that searches for a good individual (solution) fit to the given environment (problem). The GA-H stores individuals that may be transcribed by the genetic information which is discovered by the GA-P. On the other hand, the GA-P searches for useful genetic information on the GA-H, and the fitness value of each individual of the GA-P is calculated by referring to the population distribution of the individuals in the GA-H.

4.2.2 Deadlock solving optimization in GA-H

We choose n different tasks sequence $[T_1, T_2, \ldots, T_n]$ to form an individual $V_i (i = 1, 2, \ldots, 2n - 1)$. Individual V_i is expressed as a one-dimensional array $[S_1, S_2, \ldots, S_n]$, where the element S_i is equal to 0 or 1 and generated randomly during initialization. Each individual V_i represents a probable deadlock solving strategy. The initial population is expressed as $Y(0) = \{V_1(0), V_2(0), \ldots, V_M(0)\}$ and the size of $Y(0)$ is M. For the deadlock solving problem, the object cost function is:

$$C\left(V_i(k)\right) = \sum_{i=1}^{n} \text{cost}_i \times S_i \tag{4}$$

In Equation 4, $V_i(k)$ represents the individual after kth evolution; cost_i is the cost of task T_i; $C(V_i(k))$ is the cost of deadlock solving strategy.

In the GA-H, each individual in this environment has an associated fitness value that is defined as:

$$F\left(V_i(k)\right) = \sum_{i=1}^{n} \text{cost}_i - C\left(V_i(k)\right) \tag{5}$$

Individuals with higher fitness will have greater probability to be selected. Fitness function will accelerate convergence speed of the algorithm and avoid the algorithm from being trapped into a local optimum.

In the genetic algorithm, after offspring crossover and mutation, parent and offspring individuals will compete into the next population and to be selected according to the fitness. Two selected parent individuals

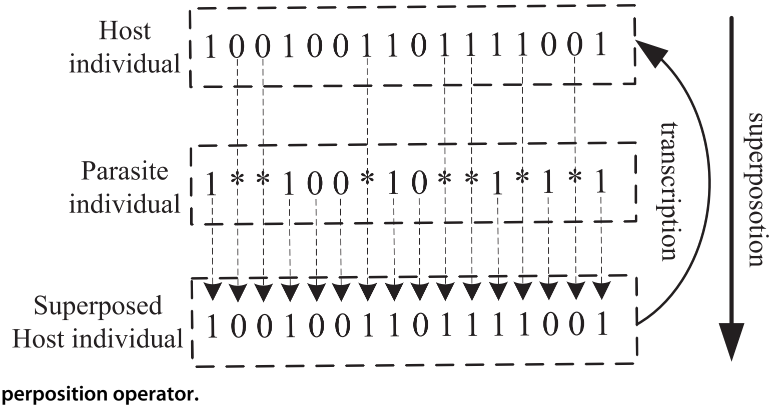

Figure 6 Mechanism of superposition operator.

may change information according to crossover probability P_c which depends on whether the genes are different in the same location. The individuals may mutate the same genes according to mutation probability P_m.

4.2.3 Deadlock solving optimization in GA-P

To improve search ability of GA-H, GA-P is adopted to inform GA-1 of the candidate subspaces to be searched for. The GA-P individuals consist of 0, 1, and * which represent genetic information in GA-H. The 'superposition' operator uses genetic information in the GA-H to calculate the fitness values of GA-2 individuals, and the 'transcription' operator propagates effective genetic information in GA-P into GA-H population. These two operators are shown in Figure 6.

In the co-evolutionary genetic algorithm, the GA-P searches for useful genetic information in GA-H. The superposition operation of each GA-P onto GA-H is carried out n times. In this process, we can describe the fitness value of jth individual of GA-P as follows: First, the GA-H individuals are denoted as $i_1, \ldots, i_n$ in and randomly decided to be superposed by GA-P individual j, and the resultant superposed GA-H individuals are denoted as $\widetilde{i_1}, \ldots, \widetilde{i_n}$. Then, the fitness evaluation of each of the superposition operations is defined as the contribution of the superposition operation to GA-H individual. The fitness evaluation can be defined as:

$$\widetilde{\nu} = max\left(0, \nu_{\widetilde{i_l}} - \nu_{i_l}\right), \tag{6}$$

where, ν_{i_l}denotes the fitness value of the i_l individual in the GA-H.

Finally, the fitness F_j of GA-P individual j is given as following:

$$F_j = \sum_{l=1}^{n} \widetilde{\nu_l} \tag{7}$$

A superposed chromosome is used to calculate the fitness value of GA-P individuals. The probability of applying the transcription operation is defined as:

$$P = \begin{cases} \widetilde{\nu_l}/\left(\nu_{\max} - \nu_{\min}\right) & \widetilde{\nu_l} > 0, \\ 0 & \text{other} \end{cases} \tag{8}$$

where $\nu_{\max}$ and $\nu_{\min}$ denote maximum and minimum fitness values in GA-H, respectively, and hence P is a value such that $0 < P < 1$.

By applying this optimization method, we can search the best deadlock solution.

Table 2 Cost of each task and the number of each resource

Tasks	Cost	Resources	Number
T_1	10	R_1	6
T_2	9	R_2	6
T_3	8	R_3	8
T_4	7	R_4	6
T_5	6	R_5	4
T_6	5	R_6	8
T_7	4		
T_8	3		

5 Experiment

In this section, we conduct experiments to validate our approaches using simulations. Our experiments are intended to: 1) verify the rationality of our proposed deadlock solving method; and 2) discuss how the model parameters affect the results. All the experiments in this paper are based on Intel CORE-i7 (2.9 GHz) machine with 8 GB RAM.

5.1 Experimental settings

In this experiment, we assume that there are eight tasks and six resources, which are involved in the vehicle sensing system. The cost of aborting a task is different. Each resource has several same kinds of factors (e.g. sensing devices, etc). Table 2 shows the cost of each task and the number of factors with each resource.

Table 3 shows the allocation of resources according to tasks.

In Table 3, each resource is held or requested by some tasks. For example, R_1 is held by T_1 and T_1 and T_8. The number of R_1 held by T_1 and T_1 and T_8 is 2, 1, and 1,

Table 3 A situation of resource allocation according to each task

Resources Task	R_1	R_2	R_3	R_4	R_5	R_6
T_1	2/0	0/3	2/0	0/0	1/0	0/0
T_2	0/0	3/0	2/0	0/1	0/0	0/0
T_3	0/0	0/2	0/0	2/0	3/0	0/0
T_4	1/0	0/0	1/0	0/3	0/0	4/0
T_5	0/0	0/2	0/0	2/0	0/2	0/0
T_6	0/0	0/0	0/2	1/0	0/0	0/3
T_7	0/1	0/0	0/2	1/0	2/0	0/0
T_8	1/0	0/0	0/2	0/0	0/0	3/0

Note: the value in this table denotes $N_{T_i \leftarrow R_j}/N_{T_i \rightarrow R_j}$. For example, the first value '2/0' that denotes two sensing devices of R_1 are hold and zero sensing devices of R_1 are requested by T_1.

Table 4 Seven strategies for deadlock solving

Main parameter	Value
Initial population size M	20
Crossover probability P_c	0.5
Mutation probability P_m	0.1
Iterations	100

respectively. R_1 is also requested by T_7, and the number of R_1 requested by T_1 is 1.

Table 4 shows the main parameters in the co-evolutionary genetic algorithm.

5.2 Experimental result and analysis

In the experiment, we let 0 stand for aborting a task while 1 stands for not aborting a task, which is explained in Section 4.1. The possible deadlock solving schemes can be expressed in the form of $[S_1\ S_2\ S_3\ S_4\ S_5\ S_6\ S_7\ S_8]$, $S_i \in \{0, 1\}$. Since there are eight tasks, the number of possible results is $2^8 - 1$. But not all the results conform to the constraints according to Equation 3. Therefore, we must exclude some individuals which do not conform to the conditions in the process of selecting the optimal deadlock solving scheme. Figure 7 shows the iterations and cost in the process of deadlock solving based on the co-evolutionary genetic algorithm.

From the experimental results, we can see that employing the genetic algorithm to solve this deadlock problem needs only six iterations, and the least cost of the optimal deadlock solving strategy is 22, which demonstrates the greatly efficiency of genetic algorithm.

In following experiments, we will investigate the impact of parameters on our method performance, including M, P_c, and P_m.

5.3 Impact of M

The parameter M is the population size. Generally, when the population size increases, the computing time or the convergence time of the algorithm will increase. In addition, the algorithm will enhance its capability of global search. To understand the impact of M, we vary the value of M as 8, 20, and 40. Figure 8 shows the fitness value and iterations in the process of deadlock solving.

From Figure 8, we can see that when the population size increases, the number of evolution iteration declines, which means that the global search ability enhances and the convergence time will fall. Particularly, these changes are not monotonic increase or decrease, but with the increase of population size, change is fluctuations or shocks.The smaller the population size, the more obvious volatility.

5.4 Impact of P_c and P_m

In order to avoid divergence or trapping in local optimum, the co-evolutionary genetic algorithm needs to adjust the P_c and P_m to keep the good individuals in a population. To study the influence of P_c and P_m on the algorithm and find the optimal parameter, we set $M = 20$. Firstly, we set $P_m = 0.1$, and set P_c as 0.1, 0.5, and 0.9; the performance

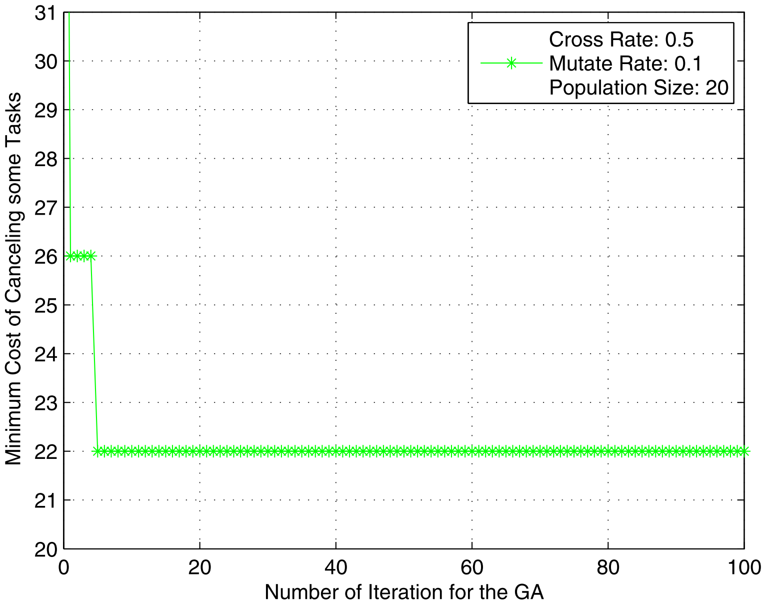

Figure 7 The cost of deadlock solving process.

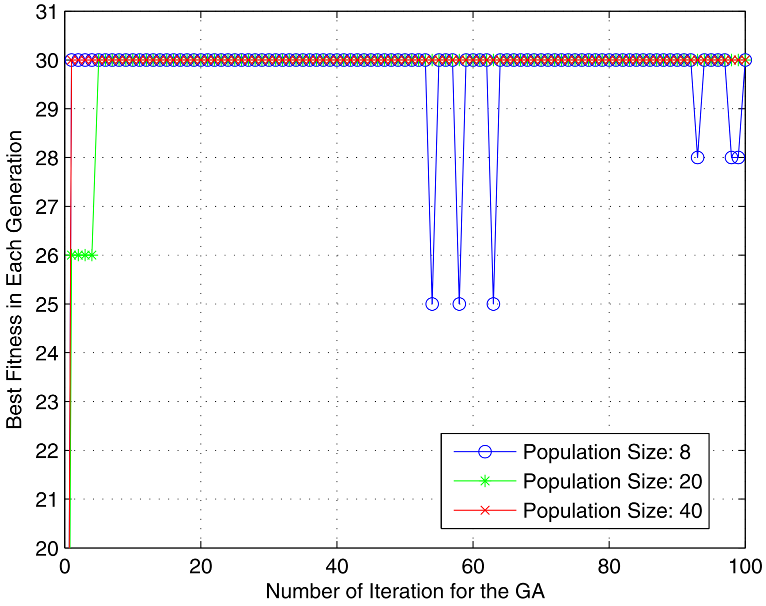

Figure 8 The impact of M.

is described in Figure 9. Secondly, we set $P_c = 0.5$, and set P_m as 0.1, 0.6, and 0.9; the performance is described in Figure 10.

Figures 9 and 10 illustrate the impact of P_c and P_m on best fitness of our model, respectively. We can see that: 1) The better P_c is 0.5 when P_c is set as 0.1, 0.5, and 0.9, while the better P_m is 0.1 when P_m is set as 0.1, 0.6, and 0.9. 2) If P_c is too small (e.g. 0.1, etc) or too large (e.g. 0.9, etc), the algorithm cannot guarantee convergence in the global optimal solution. 3) The greater the P_m, the more unstable the fitness value. If P_m is too small (e.g. 0.06, etc), the algorithm will drop into the situation of local optimum. If P_m is too large (e.g. 0.9, etc), the global search ability is also weakened. 4) Small cross rate or mutate rate will increase the iteration number of times and large cross rate or mutate rate will accelerate the calculation, but there is no guarantee that they will converge to global optimal solution.

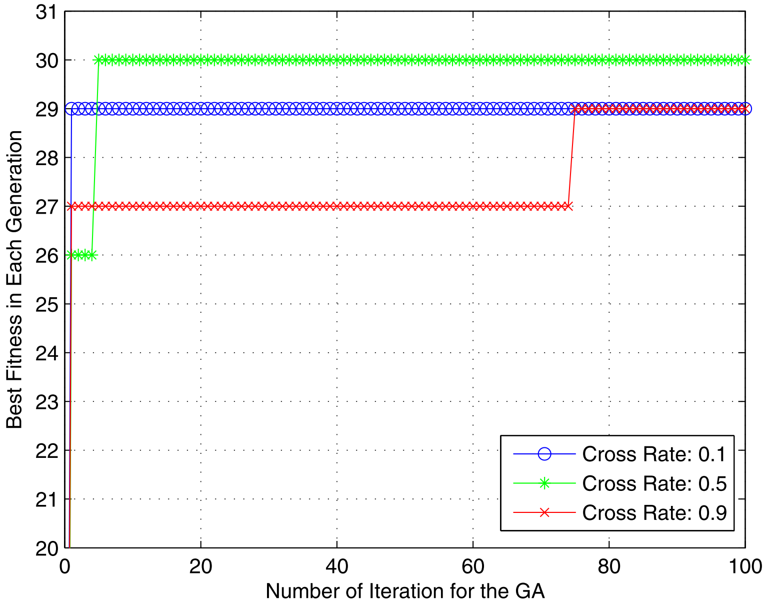

Figure 9 The impact of P_c.

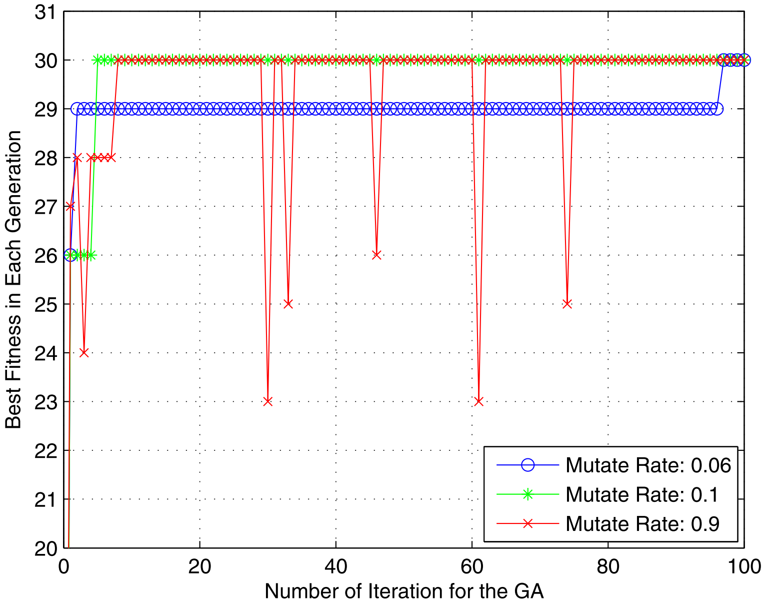

Figure 10 The impact of P_m.

6 Conclusion and future work

In this paper, we investigate the problem of deadlock in vehicle sensing systems in Internet of Vehicles. This paper proposes an efficient deadlock solving method for vehicle sensing systems. In this method, the deadlock problem is analyzed based on four deadlock occurred conditions. Tasks and resource allocation characteristics are described by mathematical expressions. Combining the quantity and cost of tasks, a deadlock solving optimization model is developed for vehicle sensing systems. To quickly solve the deadlock and to ensure the minimum cost, co-evolutionary genetic algorithm (CGA) is used to search for optimal deadlock solving strategies. Simulations are conducted and the experimental results show the efficiency of the proposed deadlock solving method based on CGA.

There are several potential future directions for our method. First, we will investigate more sophisticated ways to combine more factors (e.g. urgency degree, priority, etc.) to improve the optimization model. Second, we will propose more efficient and effective deadlock solving methods. In addition, we will also conduct more comprehensive and realistic experiments in real physical systems to evaluate our approach.

Competing interests

The authors declare that they have no competing interests.

Acknowledgments

The work described in this paper was supported by the National Natural Science Foundation of China (Grant No. 61472338 and 61332010) and the Research Grants Council of the Hong Kong Special Administrative Region, China (Grant No. CUHK 415212 of the General Research Fund).

Received: 9 September 2014 Accepted: 24 November 2014
Published: 6 December 2014

References

1. B Cheng, C Chou, C Chang, Survey on networking for internet of vehicles. J. Internet Technol. **14**, 1007–1019 (2013). doi:10.6138/JIT.2013.14.7.01
2. S Wang, C Fan, C-H Hsu, Q Sun, F Yang, A vertical handoff method via self-selection decision tree for internet of vehicles. IEEE Syst. J. **PP**, 1–10 (2014). doi:10.1109/JSYST.2014.2306210
3. Q Wang, Y Yu, Z Tang, in *Proceedings of the 2nd International Conference on Civil Engineering and Transportation*. Architecture design for intelligent vehicle computing platform based on internet of vehicles (Guilin TRANS TECH PUBLICATIONS LTD Zurich, 2012), pp. 1423–1426. doi:10.4028/www.scientific.net/AMM.253-255.1423
4. M Gerla, L Kleinrock, Vehicular networks and the future of the mobile internet. Comput. Networks **55**, 457–469. (ELSEVIER SCIENCE, Amsterdam, 2011). doi:10.1016/j.comnet.2010.10.015
5. D Leandro, V Filippo, D Marco, in *Proceeding of Automation & Test in Europe Conference & Exhibition*. Sensor networks on the car: state of the art and future challenges. Design (Grenoble IEEE Piscataway, 2011), pp. 1–6. doi:10.1109/DATE.2011.5763169
6. J Xu, G Liu, Y Gao, in *Proceeding of 2012 International Conference on Emerging Computation and Information Technologies for Education*. Vehicle operating safe state monitoring system modeling method based on automata (Springer Berlin Heidelberg, 2012), pp. 369–376
7. M Pan, G Liu, in *Proceeding of 2012 International Workshop on Information and Electronics Engineering*. Harbin. Study on vehicle operating safe state monitoring parameter and measurement model (Elsevier Amsterdam, 2012), pp. 2496–2500
8. H Yim, P Park, H Moon, J Jung, in *Ubiquitous Convergence Technology Lecture Notes in Computer Science*. An efficient sensor network architecture using open platform in vehicle environment, vol. 4412 (Springer Berlin Heidelberg, 2007), pp. 232–241
9. S Taghvaeeyan, R Rajamani, The development of vehicle position estimation algorithms based on the use of AMR sensors. IEEE Trans. Intell. Trans. Syst. **13**, 1845–1854 (2012). doi:10.1109/TITS.2012.2208189
10. S Taghvaeeyan, R Rajamani, in *Intelligent Transportation Systems*. Two-dimensional sensor system for automotive crash prediction, vol. 15 (IEEE Piscataway, 2014), pp. 178–190
11. G Georgakos, U Schlichtmann, R Schneider, in *Proceeding of the 50th ACM/EDAC/IEEE Design Automation Conference*. Austin. Reliability

challenges for electric vehicles: from devices to architecture and systems software (IEEE Piscataway, 2013), pp. 1–9

12. X Xu, C Huang, in *Proceeding of International Conference on Measurement, Instrumentation and Automation*. Guangzhou. Based on multi-sensor data fusion operating machinery velocity monitoring system (TRANS TECH PUBLICATIONS LTD Zurich, 2013), pp. 993–997
13. Wikipedia, Deadlock (2014). http://en.wikipedia.org/wiki/Deadlock
14. Z Li, N Wu, M Zhou, Deadlock control of automated manufacturing systems based on Petri nets - a literature review. IEEE Trans. Syst. Man Cybernetics Part C: Appl. Rev. **42**, 437–462 (2012). doi:10.1109/TSMCC.2011.2160626
15. Y Huang, Deadlock prevention for sequence resource allocation systems. J Inf. Sci. Eng. **23**, 215–231 (2007)
16. P Ballal, A Trivedi, F Lewis, Deadlock avoidance policy in mobile wireless sensor networks with free choice resource routing. Int. J Adv. Rob. Syst. **5**, 279–290 (2008)
17. AO Abd El-Gwad, AI Saleh, MM Abd-ElRazik, in *Proceeding of International Conference on Computer Engineering & Systems*. Cairo. A novel scheduling strategy for an efficient deadlock detection (IEEE Piscataway, 2009), pp. 579–583
18. A Aybar, A Iftar, Deadlock avoidance controller design for timed Petri nets using stretching. IEEE Syst. J. **2**, 178–188 (2008). doi:10.1109/JSYST.2008.923193
19. J-P Steghoefer, P Mandrekar, F Nafz, in *Proceeding of 23rd International Conference on Archi-tecture of Computing Systems*. On Deadlocks and fairness in self-organizing resource-flow systems (Leibniz University Hannover Hannover, 22–25 Feb 2010), pp. 87–100
20. K Xing, L Han, MC Zhou, Deadlock-free genetic scheduling algorithm for automated manufacturing systems based on deadlock control policy. IEEE Trans. Syst. Man Cybernetics, Part B (Cybernetics) **42**, 603–615 (2012). doi:10.1109/TSMCB.2011.2170678
21. R Lipset, PE Deering, RP Judd, in *Proceeding of the American Control Conference*. Albuquerque. Necessary and sufficient conditions for deadlock in manufacturing systems (IEEE Piscataway, 1997), pp. 1022–1026
22. BM Baker, MA Ayechew, A genetic algorithm for the vehicle routing problem. Comput. Operations Res. **30**, 787–800 (2003). doi:10.1016/S0305-0548(02)00051-5
23. S Saha, GARO Framework, A genetic algorithm based resource optimization for organizational efficiency. IEEE Syst. J. **7**, 889–895 (2013). doi:10.1109/JSYST.2012.2223535
24. H Handa, N Baba, O Katai, in *Proceeding of International Conference on Evolutionary Computation*. Genetic algorithm involving coevolutionary mechanism to search for effective genetic information (IEEE Piscataway, 1997), pp. 709–714. doi:10.1109/ICEC.1997.592427
25. F Abtahi, MR Meybodi, MM Ebadzadeh, R Maani, in *Proceedings of the 6th International Symposium on Intelligent Systems and Informatics*. Subotica. Learning automata-based co-evolutionary genetic algorithms for function optimization (IEEE Piscataway, 2008), pp. 1–5

doi:10.1186/1687-1499-2014-214
Cite this article as: Xu *et al.*: CGA-based deadlock solving strategies towards vehicle sensing systems. *EURASIP Journal on Wireless Communications and Networking* 2014 **2014**:214.