國 立 交 通 大 學
運 輸 科 技 與 管 理 學 系

碩 士 論 文

考慮維修需求之大眾運輸排程之研究

The Study of Public Transit Scheduling with Maintenance Requirement

研 究 生：黃嘉龍

指導教授：王晉元

中華民國九十三年七月

考慮維修需求之大眾運輸排程之研究

學生：黃嘉龍　　　　　　　　　　　　　　指導教授：王晉元

國立交通大學運輸科技與管理學系碩士班

摘　　要

由於公車的排程關係到公車的可及性、安全性以及車輛的使用期限，所以公車的排程對於大眾運輸公司來說是相當重要的。一個好的公車排程可以大眾運輸公司的營運更加有效率。然而，車輛的維修也是大眾運輸公司營運中另一個主要的要素，但在現今的營運程序中，營運前公車的預先排程通常未考慮車輛的維修需求。預排的公車排程必須透過調度員的調整才能符合維修需求。

近幾年來，螞蟻族群演算法被應用在許多的實例上，並且螞蟻族群演算法通常比其他演算法有較佳的結果或效率，所以本研究提出一個以螞蟻族群系統(ACS)為基礎之演算法來求解考慮維修需求之大眾運輸排程的問題。

在本研究所提出的演算法中，共有 5 項重要的課題，分別為(1)網路設計，(2)狀態轉換規則，(3)路徑構建規則，(4)費洛蒙區域更新規則，以及(5)費洛蒙全域更新規則，並且根據這些規則來來求解考慮維修需求之大眾運輸排程的問題。

為證明演算法的實用性，本研究從一家大眾運輸公司取得實際的營運資料，並且產生實際的測試範例來進行測試。經過測試結果證實，本研究所提出的演算法可以解決實務上的問題，並解同時產製公車的營運排班以及維修排班結果。

關鍵字：大眾運輸排程，螞蟻族群演算法，維修需求，費洛蒙

The Study of Public Transit Scheduling with Maintenance Requirement

Student : Chia-Lung Huang　　　　　　Advisor : Jin-Yuan Wang

Department of Transportation Technology and Management

National Chiao Tung University

Abstract

Because the bus scheduling relates to reliability, safety and vehicle life, the bus schedule is very important to the public transit company. A good bus scheduling can make the operation of the public transit company more efficient. Vehicle maintenance is another major component in public transit operations. But the current operation procedures of bus scheduling is often established before considering the vehicle maintenance requirement. The schedule must be modified by dispatchers to fulfill the requirement.

In recent years, the ant colony algorithm are used in many cases. The ant colony algorithm often has better solutions or efficiency better than other algorithms do. So we propose an ant colony system (ACS) based algorithm to solve this problem of public transit scheduling with maintenance requirement.

The five major issues embedded in our algorithm are 1.network designing, 2.the state transition rule, 3.the route building rule, 4.the pheromone local updating rule and 5.the pheromone global updating rule. According to these rules, we will solve the problem of public transit scheduling with maintenance requirement.

In order to prove this algorithm, we obtain real operation data from a public transit company, and we test these small real samples through generating those testing examples. The testing results confirm that this algorithm can successfully solve the problem of a real operation and meanwhile, it also generates the schedules of the buses operation and maintenance.

Keyword: Public Transit Scheduling, Ant Colony Algorithm, Maintenance Requirement, Pheromone.

誌　　謝

本論文得以順利完成，首先要感謝王晉元老師的細心指導。在論文的撰寫過程中，每當學生遇到難題時，老師均不辭辛勞的給予指導以及指引方向。然而由於論文是用英文來撰寫，因此老師更是花更多的時間與精神在我們五位研究生身上。除了研究與課業的指導外，老師在生活上以及處理事情的態度方法上也給予相當多的經驗傳承。謝謝老師這兩年來的教導與照顧，謹此致上最深的敬意。

在論文口試期間，感謝中華大學運輸科技與物流管理學系 蘇昭銘老師以及張靖老師在論文進行過程中給予相當多寶貴的意見以及指導，最後在論文口試時惠賜卓見及不吝指正，使本論文正臻完備；在論文進度審查期間，也承系上 韓復華老師以及 卓訓榮老師費心審閱並給予指導。

在這兩年研究生的生活中，感謝至康(大師兄)在生活上、學業上各方面的指導，而每次都可在大師兄的生活經歷上發現許多刺激好玩的事^^Y；感謝猴子、knight、欣潔和阿信幾個同甘共苦的戰友一起度過在實驗室的生活，我永遠記得每次在meeting前一個晚上大家都在等待 meeting time 的 mail 的那種情況;感謝建名學長這麼多年的陪伴(從大學到研究所)，讓我感受到人真的不可做壞事，因為會有報應:P；感謝家盛、駿逸兩位學長在程式上的指點，也從家盛的身上得知畜生之道 XD；感謝家族學妹怡君、小昱平時在學校生活上的相互照應及關心；感謝實驗室學弟妹 hoho、彥佑、嘉英平日在實驗室勁爆的談話，使得生活多了許多的樂趣，我也永遠記得被嘉英莫名陷害的情況(嘉英妳給我記住…)。在此，祝實驗室的學長、學弟妹都能順利的畢業。

感謝室友阿達、酷璋、董鵝、ㄟ蕭陪我一起在宿舍打拼打屁，有了你們使我在宿舍的生活多采多姿，一點都不無聊；感謝高中好友瑞敏、瑩盈、亞玫、思賢、虹年平日的關心與關懷。

感謝研究所的同學依潔、惠玉、小宇、梅子、大乃、文秀、大象、阿貴、小渣、佳琴、偉哲、小紅、士偉、則言、韻璇、起豪、苑綾、大中……等，有了你們使我的研究所的生活更加的充實愉快。

最後感謝一直默默支持我的家人，有你們的關心、叮嚀與照顧，才能讓我全心的投入學校的研究工作。誠心的感謝在這段期間關心、支持我的人，也與大家一同分享論文完成時的喜悅，感謝大家。

黃嘉龍

新竹交大

2004/7/26

Content

Table List

Figure List

Chapter1 Introduction

1.1 Motivation

A public transit company operational planning process includes four basic sequential tasks: (1) network route design, (2) setting timetable, (3) scheduling vehicles to trips, and (4) drivers assignment. It is desirable for all tasks to be planned simultaneously in order to optimize productivity and efficiency. However, due to the complexity of the problem, transit planning is performed in a sequential manner in which the outcome of one step is fed as an input to the next step.

Because the bus scheduling relates to reliability, safety and vehicle life, the bus scheduling is very important to the public transit company. A good bus scheduling can make the operation of the public transit company more efficient.

Vehicle maintenance is another major component in public transit operations. But the current operation procedures of bus scheduling is often established before considering the vehicle maintenance requirement. The schedule must be modified by dispatchers to fulfill the requirement. Hence, the development of a bus scheduling algorithm, which takes vehicle maintenance into consideration to assist human dispatchers, would certainly be very useful.

1.2 Objective

The objective of this research is to develop an algorithm for bus operation scheduling which takes vehicle maintenance requirement into consideration. Based on the given vehicle maintenance regulations, an effective algorithm is proposed. Computational experiments are conducted for testing purposes.

1.3 Scope

We focus on daily vehicles scheduling problem of the public transit companies, we also take vehicle maintenance requirement into consideration. The algorithm that we propose is applicable to both the urban buses and inter-city buses.

1.4 Study Flowchart

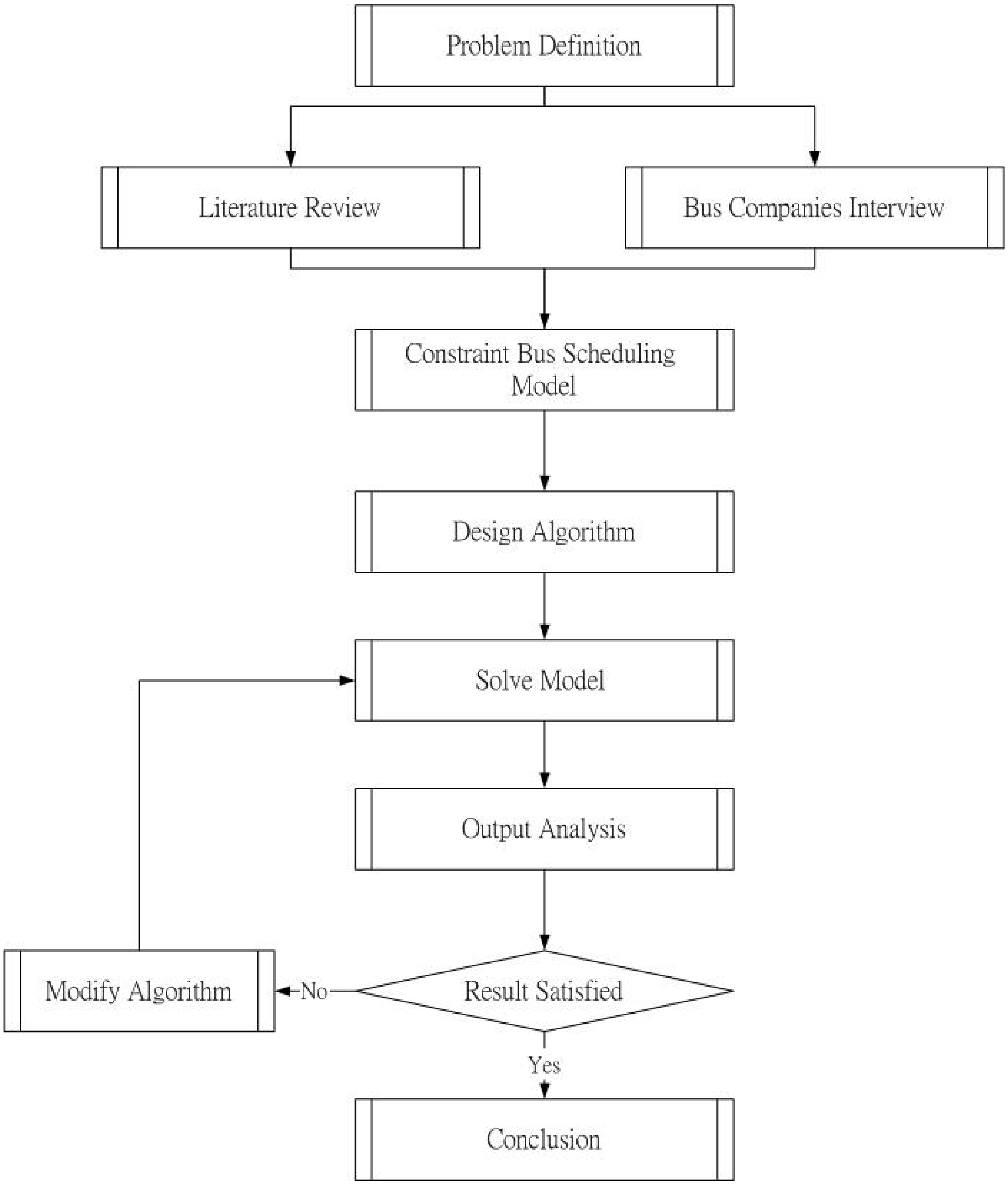

Figure 1-1 Study Flowchart

First, we define the problem. After realizing the problem we encounter, we will review relevant literatures, and interview the bus companies to know current operation procedures.

We will then formulate this vehicle scheduling problem. Thereafter, a solution algorithm is proposed to solve the model.

Finally, we use some data to test our module. If the result is not satisfied, we will keep modifying our algorithm until the result satisfied the actual requirement.

Chapter2 Literature Review

We review the relevant literature in this chapter. We categorize these literatures into three aspects. There are the problem of scheduling bus maintenance activities, the problem of assigning planes to fights, and the problem of planning airline maintenance manpower. In the last section, we introduce the ant colony system algorithm.

2.1 The Problem of Scheduling Bus Maintenance Activities

Haghani and Shafahi(2002)[1] presented a mathematical programming approach to the problem of scheduling bus maintenance activities. This approach took as input a given daily operating schedule for all buses assigned to a depot along with available maintenance resources. It, then, attempted to design daily inspection and maintenance schedules for the buses that are due for inspection so as to minimize the interruptions in the daily bus operating schedule, and maximize the utilization of the maintenance facilities. Three integer programming formulations are presented and different properties of the problem are discussed. Several heuristic methods are presented and tested. Some of these procedures produce very close to optimal solutions very efficiently. In some cases, the computational times required to obtain these solutions are less than 1% of the computational time required for the conventional branch and bound algorithm. Several small examples are offered and the computational results of solving the problem for an actual, 181-bus transit property are reported.

2.2 The Problem of Assigning Planes to Fights

Moudani and Mora-Camino(2000)[2] discussed the problems of assigning planes to fights and of fleet maintenance operations scheduling are considered in this paper.

While recent approaches make use of artificial intelligence techniques running on mainframe computers to solve combinatorial optimization problems for nominal operations, a dynamic approach is proposed here to face on-line operation conditions. The proposed solution mixes a Dynamic Programming approach (to cope with the fleet assignment problem) and a heuristic technique (to solve the embedded maintenance schedule problem). When applied to a medium charter airline, this approach shows acceptability characteristics for operational staffs, while providing efficient solutions. The proposed solution scheme can be considered as the basis for the development of an on-line decision support system for fleet operations management within airlines.

2.3 The Problem of Planning Airline Maintenance Manpower

Yang and Chen(2001)[3] presented an airline maintenance manpower planning model with flexible strategies. They used mathematical programs and computer algorithms to develop suitable models and solution methods, in order to help airlines efficiently and effectively plan their maintenance schedules and manpower supplies, which are then useful for downstream maintenance crew assignments. Because the problem size of the mixed integer programming model is expected to be huge, they developed a heuristic solution framework to solve the problem. The framework is divided into three stages. In the first two stages, two integer programs are formulated respectively. A mixed integer program is formulated in the third stage. The first model is used to determine the best shift plans. The second model is used to solve the maintenance manpower supply problem for each aircraft type. The third model helps simultaneously solve the maintenance manpower supply problem for mixed aircraft types, based on the fact that maintenance crew members are practically qualified for repairing different aircraft types in a work shift. They solved all stages by using the mathematical programming solver, CPLEX, and other self-developed computer programs. Finally, to evaluate the models and solution algorithms developed in the research, we perform a case study using the operating data from a major Taiwan airline. The results show that the models and the solution methods are useful.

2.4 Introduction of Ant Colony Optimization Algorithm[4]

The Ant System, introduced by Colorni, Dorigo and Maniezzo is a new distributed meta-heuristic for hard combinatorial optimization problems and was first applied on the well known Traveling Salesman Problem(TSP).

Observation on real ants searching for food were the inspiration to imitate the behavior of ant colonies for solving combinatorial optimization problems. Real ants are able to communicate information concerning food sources via an aromatic essence, called pheromone. They mark the path they walk on by laying down pheromone in a quantity that depends on the length of the path and the quality of discovered food source. Other ants can observe the pheromone trail and are attracted to follow it. Thus, the path will be marked again and will therefore attract more ant. The pheromone trail on paths leading to rich food sources close to the nest will be more frequented and will therefore grow faster.

The described behavior of real ant colony can be used to solve combinatorial optimization problems by simulation: artificial ants searching the solution space simulate real ants searching their environment, the objective values correspond to the quality of the food sources and adaptive memory corresponds to the pheromone trails. In addition, the artificial ants are equipped with a local heuristic function to guide their search through the set of feasible solutions.

Ant system is the progenitor of all our research efforts with ant algorithms and was first applied to the TSP, which is defined in Figure 2-1.

TSP

Let V = {a, ⋯, z} be a set of cities, A = {(r, s) : r, s ∈ V} be the edge set, and δ(r, s) = δ(s, r) be a cost measure associated with edge (r, s) ∈ A.

The TSP is the problem of finding a minimal cost closed tour that visits each city once.

In the case cities r ∈ V are given by their coordinates (x_r, y_r) and δ(r, s) is the Euclidean distance between r and s, then we have an Euclidean TSP.

ATSP

If δ(r, s) ≠ δ(s, r) for at least some (r, s) then the TSP becomes an asymmetric TSP(ATSP).

Figure 2-1 The traveling salesman problem

Ant system utilizes a graph representation which is the same as that defined in Fig. 2, augmented as follows: in addition to the cost measure $\delta(r,s)$, each edge *(r,s)* has also a desirability measure $\tau(r,s)$, called pheromone, which is updated at run time by artificial ants (ants for short). When ant system is applied to symmetric instances of the TSP, $\tau(r,s) = \tau(s,r)$, but when it is applied to asymmetric instances it is possible that $\tau(r,s) \neq \tau(s,r)$.

Informally, ant system works as follows. Each ant generates a complete tour by choosing the cities according to a probabilistic state transition rule; ants prefer to move to cities which are connected by short edges with a high amount of pheromone. Once all ants have completed their tours a global pheromone updating rule (global updating rule, for short) is applied; a fraction of the pheromone evaporates on all edges (edges that are not refreshed become less desirable), and then each ant deposits an amount of pheromone on edges which belong to its tour in proportion to how short its tour was (in other words, edges which belong to many short tours are the edges which receive the greater amount of pheromone). The process is then iterated.

The state transition rule used by ant system, called a random-proportional rule, is given by (2-1), which gives the probability with which ant k in city r chooses to move to the city s

$$p_k(r,s)=\begin{cases}\dfrac{[\tau(r,s)]\cdot[\eta(r,s)]^{\beta}}{\sum_{u\in J_k}[\tau(r,u)]\cdot[\eta(r,u)]^{\beta}}, & if \quad s\in J_k(r)\\ 0, & otherwise\end{cases} \tag{2-1}$$

where τ is the pheromone, $\eta = 1/\delta$ is the inverse of the distance $\delta(r,s)$, $J_k(r)$ is the set of cities that remain to be visited by ant k positioned on city r (to make the solution feasible), and β is a parameter which determines the relative importance of pheromone versus distance ($\beta>0$).

In (2-1) we multiply the pheromone on edge *(r,s)* by the corresponding heuristic value $\eta(r,s)$. In this way we favor the choice of edges which are shorter and which have a greater amount of pheromone.

In ant system, the global updating rule is implemented as follows. Once all ants have built their tours, pheromone is updated on all edges according to

$$\tau(r,s)\leftarrow(1-\alpha)\cdot\tau(r,s)+\sum_{k=1}^{m}\Delta\tau_k(r,s) \tag{2-2}$$

where

$$\Delta\tau_k(r,s)=\begin{cases}\dfrac{1}{L_k}, & if\ (r,s)\in tour\ done\ by\ ant\ k\\ 0, & otherwise\end{cases}$$

$0<\alpha<1$ is a pheromone decay parameter, L_k is the length of the tour performed by ant *k*, and *m* is the number of ants.

Pheromone updating is intended to allocate a greater amount of pheromone to shorter tours. The pheromone updating formula was meant to simulate the change in the amount of pheromone due to both the addition of new pheromone deposited by ants on the visited edges and to pheromone evaporation.

Pheromone placed on the edges plays the role of a distributed long-term memory: this memory is not stored locally within the individual ants, but is distributed on the edges of the graph. This allows an indirect form of communication called stigmergy.

The interested reader will find a full description of ant system.

Although ant system was useful for discovering good or optimal solutions for small TSP's (up to 30 cities), the time required to find such results made it infeasible for larger problems. We devised three main changes to improve its performance which led to the definition of the ACS, presented in the next section.

The ACS differs from the previous ant system because of three main aspects: i) the state transition rule provides a direct way to balance between exploration of new edges and exploitation of a priori and accumulated knowledge about the problem, ii) the global updating rule is applied only to edges which belong to the best ant tour, and iii) while ants construct a solution a local pheromone updating rule (local updating rule, for short) is applied.

Informally, the ACS works as follows: ants are initially positioned on cities chosen according to some initialization rule (e.g., randomly). Each ant builds a tour (i.e., a feasible solution to the TSP) by repeatedly applying a stochastic greedy rule (the state transition rule). While constructing its tour, an ant also modifies the amount of pheromone on the visited edges by applying the local updating rule. Once all ants have terminated their tour, the amount of pheromone on edges is modified again (by applying the global updating rule). As was the case in ant system, ants are guided, in building their tours, by both heuristic information (they prefer to choose short edges) and by pheromone information. An edge with a high amount of pheromone is a very desirable choice. The pheromone updating rules are designed so that they tend to give more pheromone to edges which should be visited by ants. The ACS algorithm is reported in Figure 2-2. In the following we discuss the state transition rule, the global updating rule, and the local updating rule.

```
Initialize
Loop /* at this level each loop is called an iteration */
      Each ant is positioned on a starting node
      Loop /* at this level each loop is called a step */
            Each ant applies a state transition rule to incrementally
            build a solution and a local pheromone updating rule
      Until all ants have built a complete solution
      A global pheromone updating rule is applied
Until End_condition
```

Figure 2-2 The ACS algorithm

2.4.1 ACS State Transition Rule

In the ACS the state transition rule is as follows: an ant positioned on node chooses the city to move to by applying the rule given by (2-3)

$$s = \begin{cases} \arg \max_{u \in J_k} \{[\tau(r,u)] \cdot [\eta(r,u)]^{\beta}\}, & if \quad q \le q_0 \quad (\exp lotitation) \\ S, \quad otherwise & (biased \quad \exp loration) \end{cases} \tag{2-3}$$

where q is a random number uniformly distributed in [0…1], q_0 is a parameter ($0 \leqq q_0 \leqq 1$), and S is a random variable selected according to the probability distribution given in (2-1).

The state transition rule resulting from (2-3) and (2-1) is called pseudo-random-proportional rule. This state transition rule, as with the previous random-proportional rule, favors transitions toward nodes connected by short edges and with a large amount of pheromone. The parameter q_0 determines the relative importance of exploitation versus exploration: every time an ant in city r has to choose a city s to move to, it samples a random number $0 \leqq q \leqq 1$. If $q \leqq q_0$ then the best edge, according to (2-3), is chosen (exploitation), otherwise an edge is chosen according to (2-1) (biased exploration).

2.4.2 ACS Global Updating Rule

In ACS only the globally best ant (i.e., the ant which constructed the shortest tour from the beginning of the trial) is allowed to deposit pheromone. This choice, together with the use of the pseudo-random-proportional rule, is intended to make the search more directed: ants search in a neighborhood of the best tour found up to the current iteration of the algorithm. Global updating is performed after all ants have completed their tours. The pheromone level is updated by applying the global updating rule of (2-4).

$$\tau(r,s) \leftarrow (1-\alpha)\cdot\tau(r,s) + \alpha\cdot\Delta\tau(r,s) \tag{2-4}$$

where

$$\Delta\tau(r,s) = \begin{cases} (L_{gb})^{-1}, & if\ (r,s) \in global-best-tour \\ 0, & otherwise \end{cases}$$

0< α <1 is the pheromone decay parameter, and L_{gb} is the length of the globally best tour from the beginning of the trial. As was the case in ant system, global updating is intended to provide a greater amount of pheromone to shorter tours. Equation (2-4) dictates that only those edges belonging to the globally best tour will receive reinforcement. We also tested another type of global updating rule, called iteration-best, as opposed to the above called global-best, which instead used L_{ib} (the length of the best tour in the current iteration of the trial), in (2-4). Also, with iteration-best the edges which receive reinforcement are those belonging to the best tour of the current iteration. Experiments have shown that the difference between the two schemes is minimal, with a slight preference for global-best, which is therefore used in the following experiments.

2.4.3 ACS Local Updating Rule

While building a solution (i.e., a tour) of the TSP, ants visit edges and change their pheromone level by applying the local updating rule of (2-5)

$$\tau(r,s) \leftarrow (1-\rho)\cdot\tau(r,s) + \rho\cdot\Delta\tau(r,s) \tag{2-5}$$

where 0< ρ <1is a parameter.

Chapter3 The Algorithm of Public Transit Scheduling with Maintenance Requirement

In this chapter we describe real operation procedures of public transit scheduling with maintenance requirement and propose an ant colony system (ACS) based algorithm to solve this problem. The detail of this algorithm is explained in the following sections.

3.1 The Operation Procedures of Public Transit Scheduling with Maintenance Requirement

In real operation, we can divide the bus maintenance into tree categories: daily inspection, preventive (periodic) maintenance, and emergency maintenance. This type of inspection is defined as a general check by the drivers or other personnel at the beginning or at the end of a working day. Thus, this inspection does not have any scheduling requirement.

Emergency maintenance is required when buses have unexpected breakdowns. There is no way of knowing in advance whether or not a bus will break down on a particular day. Breakdown of a bus during service is the most costly and disruptive situation that results in a road call, and requires that the maintenance manager dispatches a mechanic to the site to determine the cause and the nature of the breakdown, and fix it if possible. This is a significant drain on the maintenance resources because a job in progress must be put on-hold to dispatch a mechanic to the breakdown site. In most cases, the bus is towed back to the facility. Depending on the nature of breakdown, it is either fixed at the site, or the bus is sent to a central maintenance depot for major repair work. When the bus is to be fixed at the facility, it can be considered as yet another bus that requires maintenance, and this maintenance activity can be scheduled with the other regular maintenance activities. If the bus is sent to another facility, there is no need for it to be considered in scheduling.

Preventive maintenance is the regular inspection of buses in pre-specified mileage or time interval. During these inspections, certain parts of buses are replaced or maintained. If other components are identified that deemed near failure, they are replaced too. In this research, we discuss preventive maintenance for public transit scheduling with maintenance requirement.

Figure 3-1 is the current operation procedure of public transit scheduling with maintenance requirement.

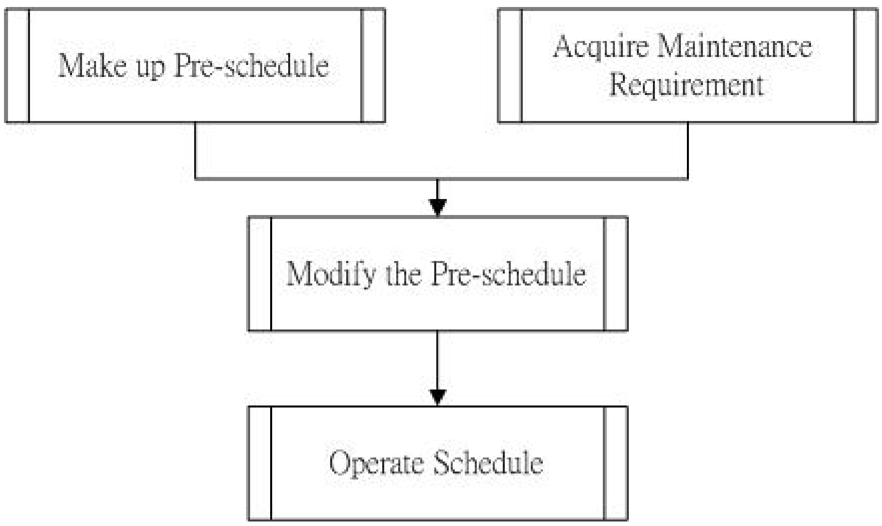

Figure 3-1 The operation schedule production traditional

The pre-schedule of buses is made before the daily operation. The dispatcher also require maintenance requirement from the maintain factory base on the mileage or time interval of each bus. By putting this two information into consideration, the dispatcher modifies the pre-schedule to accommodate the maintenance requirement. Several issues are raised during this operation.

1. Depend on the dispatcher excessively: In real operation, the dispatcher is very important, and his workload is very heavy. Thus, the dispatcher may not consider comprehensively to modify the pre-schedule to accommodate the

maintenance requirement in an efficient and effective way. So the final schedule is not efficient.

2. Cause the impact of the schedule: This modification causes unexpected changes of pre-schedule, which result in a certain degree of confusion of drivers.
3. The final schedule for the bus is not effective: Because the final schedule for buses maybe have too much idle time.

So we provide a mechanism to address these issues.

3.2 Problem Definition

The public transit scheduling with maintenance requirement can be represented by a weighted directed graph $G = (N,A,d)$ where $N = \{n_0,n_1,n_2,\ldots,n_n\}$ is a set of nodes and $A= \{(n_i,n_j) : i \neq j\}$ is a set of arcs. The n_0 denotes the start node, the other nodes represent tasks and maintenances. The nonnegative weights d_{ij}, which are associated with each arc (n_i,n_j), represent the idle time between n_i and n_j. Each task n_i records its route trip distance and trip time. The aim is to find minimum cost (minimum idle time) vehicle services tasks where

- Every task is serviced exactly once by exactly one vehicle.
- All vehicle schedules begin and end at the start node.
- The traveled distance does not exceed the vehicle maintenance requirement for each vehicle.

The object of this problem is minimum vehicle idle time and the number of vehicles. The object function is (3-1)

$$Min \sum t_l + V_{um} \cdot V_C \qquad l \in L \tag{3-1}$$

where L is the set of arcs that two tasks are serviced by one vehicle continuously, V_{um} is the number of vehicles, and V_C is the cost of each vehicle.

The problem of public transit scheduling with maintenance requirement is a very complicated combinatorial optimization problem like Vehicle Routing Problem (VRP) and Traveling Salesman Problem (TSP). In recent years, the ant colony algorithm are used in many cases. The ant colony algorithm often has better solutions or efficiency better than other algorithms do(ex. GA, SA)[4][8]. So we propose an ant colony system (ACS) based algorithm to solve this problem.

3.3 Characteristics of Artificial Ants

There are some characteristics of artificial ants in this research. Those characteristics are listed as follow:

1. Artificial ants can remember the visited nodes.
2. When the candidate node is ϕ, the ant must turn to the start node.
3. Artificial ants can remember the traveled mileage.
4. Artificial ants can remember the maintenance requirement travel mileage of vehicles.
5. When each ant visits all nodes, it represents a schedule of all vehicles.

3.4 Solution Algorithm Flowchart

The flowchart of solution algorithm is shown in figure3-2. The steps of the algorithm are listed as follows:

Step 1: We collect the real data from the bus company. The data include duties, vehicle number, the route distance and the maintenance requirement rules.

Step 2: We use duties to build the network and set the parameters. The parameters include the number of ants, the count of colonies, pheromone decay parameter and determining the relative importance of pheromone parameter.

Step 3: In the initiation, we position each ant on earliest *m* tasks. The *m* is the number of ants.

Step 4: We design the state transition rule, and use the rule to build tours. The rule must consider the maintenance requirement and the idle time of vehicles.

Step 5: We design the pheromone local updating rule. Ants visit edges and change their pheromone level by applying the local updating rule.

Step 6: If all ants of the colony have completed their tours. Then we find the best tour.

Step 7: After finding the best tour, we apply the pheromone global updating rule. Only the globally best ant is allowed to deposit pheromone. Global updating is performed after all ants have completed their tours. The pheromone level is updated by applying the global updating rule.

Step 8: If the number of replication is reached, algorithm stop and outputs the best solution.

We describe the five major issues embedded in the above algorithm in the following sections. There are network designing (section 3.4.1), the state transition rule (section 3.4.2), the route building rule (section 3.4.3), the pheromone local updating rule (section 3.4.4), and the pheromone global updating rule (section 3.4.5).

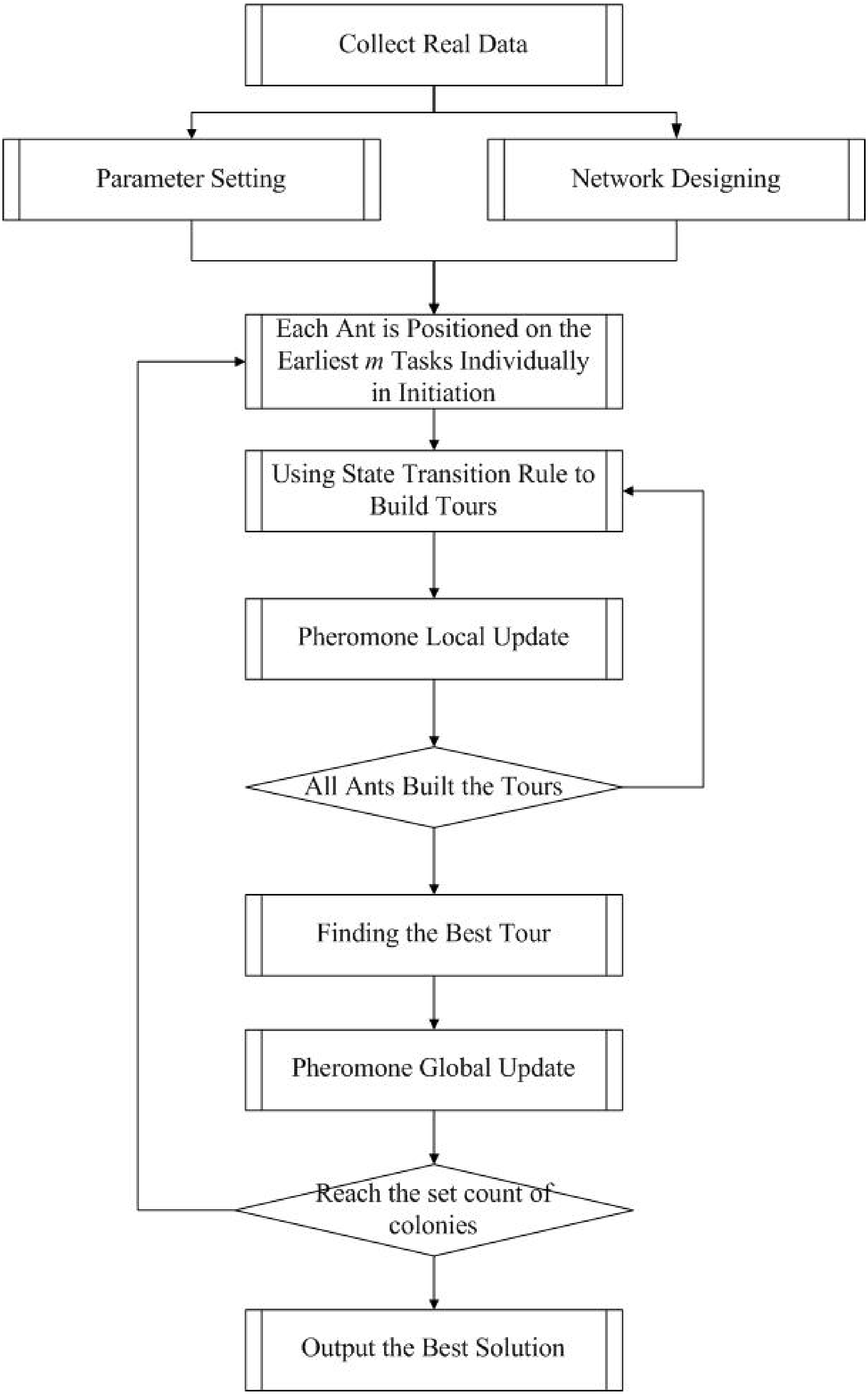

Figure 3-2 Solution algorithm flowchart

3.4.1 Network Designing

We define a network for vehicle scheduling. The nodes include the start node, maintenance nodes and task nodes. The arcs are annotated with the costs to serve them.

Let G indicates the whole vehicle scheduling network, N indicates the set of nodes and A indicates the set of arcs. Thus, $G = (N,A)$.

There are four kinds of arcs. The first one is from the start node to any task node, the second one is from any task node to another task node, the third one is from any task node to the maintenance node, the fourth one is from maintenance node to any task node. Each arc means a feasible connection of two nodes. It is considered a feasible connection if one node which the task begin time late 10 minutes more than another node.

In Figure 3-3, each task node has four numbers associated. The first number is the serial number of task. The second number is the starting time to carry out the task. The third number is the ending time of that task (e.g. the starting time and ending time for the task 1 is 6:00 and 7:00). The fourth number is the trip distance of that task. Each maintenance node has four numbers associated. The first number is maintenance type. The second number is the serial number of maintenance task. The third number is the starting time to carry out the maintenance. The fourth number is the ending time of that maintenance. The start node has one number associated. The number is the starting time of the start node.

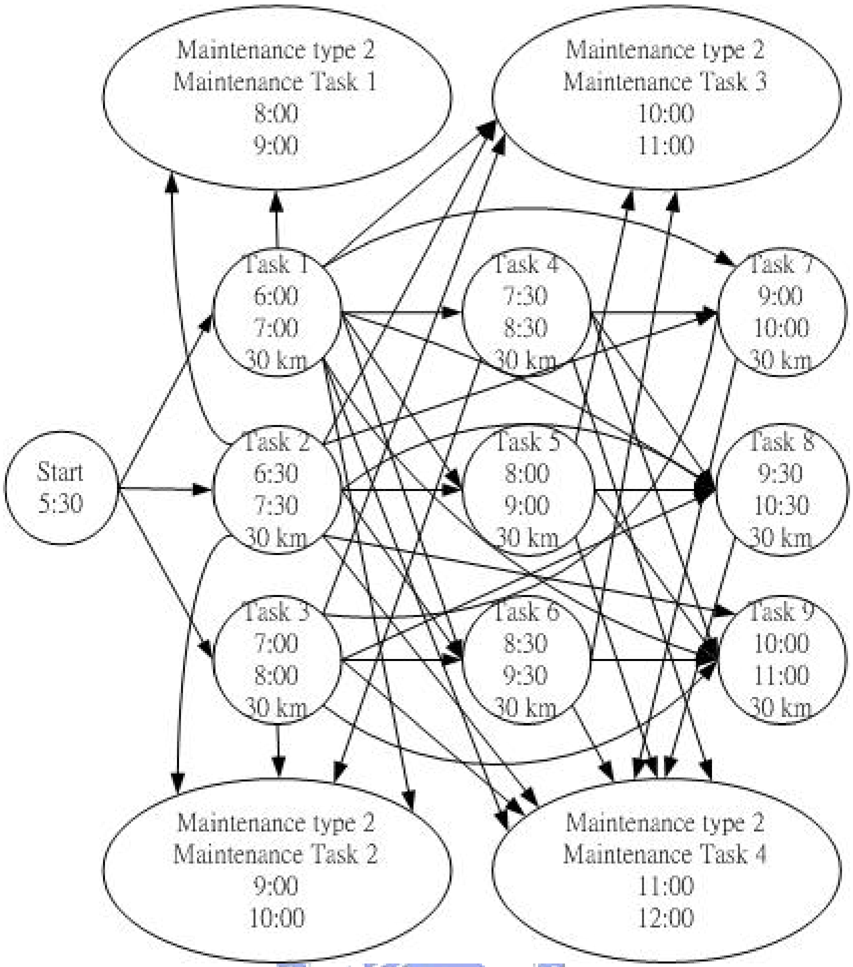

Figure 3-3 Example of a vehicle scheduling network

3.4.2 State Transition Rule

In the state transition rule, an ant positioned on node r chooses the node s to move to by following rule (3-2)

$$s = \begin{cases} \arg \max_{u \in J_k(r)} \{[\tau(r,u)] \cdot [\eta(r,u)]^{\beta}\}, & if \quad R_r^v < R_m^v \quad and \quad q \le q_0 \\ S, \quad if \quad R_r^v < R_m^v \quad and \quad q > q_0 & \\ \arg \max_{u \in M_k(r)} \{[\tau(r,u)] \cdot [\eta(r,u)]^{\beta}\}, & if \quad R_r^v \ge R_m^v \end{cases} \tag{3-2}$$

$$S = \begin{cases} \dfrac{[\tau(r,s)] \cdot [\eta(r,s)]^{\beta}}{\sum_{u \in J_k(r)} [\tau(r,u)] \cdot [\eta(r,u)]^{\beta}}, & if \ s \in J_k(r) \\ 0\,, & otherwise \end{cases} \tag{3-3}$$

where q is a random number uniformly distributed in [0..1], q_0 is a parameter (0≦q_0 ≦0), S is a random variable selected according to the probability distribution given in

(3-3), τ is the level of pheromone, $\eta = 1/t$ is the inverse of the idle time $t(r,s)$, $J_k(r)$ is the set of task nodes that remain to be visited by ant k positioned on node r, $M_k(r)$ is the set of maintenance nodes that remain to be visited by ant k positioned on node r, R_r^v is the traveled miles for vehicle v, R_m^v is maintenance requirement travel miles for vehicle v.

The parameter q_0 determines the relative importance of exploitation versus exploration: every time an ant in city r has to choose a city s to move to, it samples a random number $0 \leqq q \leqq 1$. If $q \leqq q_0$ and $R_r^v < R_m^v$ (The ant do not move to maintenance node.) then the best arc, according to (3-2), is chosen (exploitation). If $q > q_0$ and $R_r^v < R_m^v$ an arc is chosen according to (3-3) (biased exploration). If $R_r^v \geqq R_m^v$ the ant must move to maintenance node then the arc, according to (3-2), is chosen.

3.4.3 Route Building Rule

Each ant start at the start node, and build the route according to state transition rule. When the candidate list is ϕ ($J_k(r)=\phi$ or $M_k(r)=\phi$), the ant must return to start node, and the ant represents the different vehicle. While all task nodes are visited by an ant, the routes are built.

3.4.4 Pheromone Local Updating Rule

While building a solution, ants visit edges and change their pheromone levels by applying the local updating rule of (3-4)

$$\tau(r,s) \leftarrow (1-\rho)\cdot\tau(r,s) + \rho\cdot\Delta\tau_0 \qquad (3\text{-}4)$$

where $0<\rho<1$ is a pheromone decay parameter.

3.4.5 Pheromone Global Updating Rule

In ACS only the globally best ant is allowed to deposit pheromone. Global updating is performed after all ants have completed their tours. The pheromone level is updated by applying the global updating rule of (3-5)

$$\tau(r,s) \leftarrow (1-\alpha)\cdot\tau(r,s) + \alpha\cdot\Delta\tau(r,s) \tag{3-5}$$

where $0 < \alpha < 1$ is a pheromone decay parameter.

Chapter4 Model Testing

In this chapter we use a simple example to test the algorithm, and we obtain real operation data from public transit companies to generate the real example to test the algorithm.

4.1 Simple Test

We have the timetable shown in Table 4-1 in this simple example. We suppose the each run of route1 needs 1 hour, the each run of route 2 needs 45 minutes, and each run needs to back to the depot. Thus, we design the network based on this timetable, and the network is shown in Figure 4-1. We set $\beta = 1$, $\rho = 0.5$, $\alpha = 0.9$, $\tau_0 = 1$, $q_0 = 0.9$, $\Delta\tau(r,s) = 1$ and $\Delta\tau_0 = 1$. The number of ants used is 2($m = 2$). The count of colonies is 2($n = 2$). We assume the number of vehicles is 5, and the traveled mileage are $R_r^1 = 2000$ km, $R_r^2 = 2910$ km, $R_r^3 = 1000$ km, $R_r^4 = 2500$ km and $R_r^5 = 1000$, and the maintenance requirement travel mileage is $R_m^1 = R_m^2 = R_m^3 = R_m^4 = R_m^5 = 3000$ km, and the cost of each vehicle is 1000(V_C=1000). The route 1 is 100 km long and route 2 is 25 km long. Maintenance requires 2 hours to finish. The steps of solution are listed below:

Table 4-1 Simple of timetable

Route 1		Route 2
6:00	9:00	6:15
6:30	9:30	7:15
7:00	10:00	8:15
7:30	10:30	9:15
8:00	11:00	10:15
8:30		

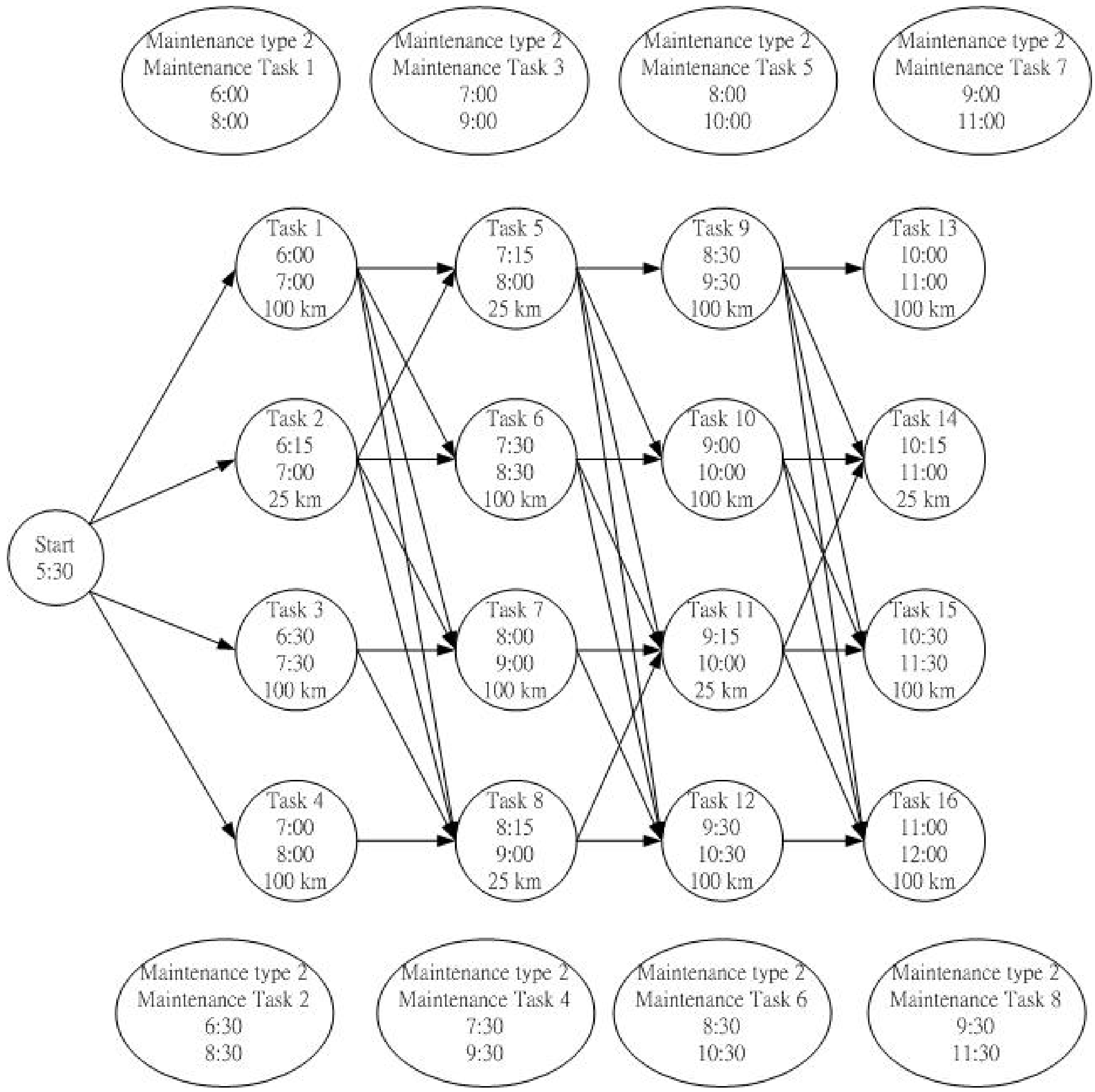

Figure 4-1 Simple of network

Iteration 1:

Initial cost matrix: the number in each element is the idle time between any two tasks.

(node 0 is the start node)

t(r,s)=

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	--	30	45	60	90	105	120	150	165	180	210	225	240	270	285	300	330
1	--	--	--	--	--	15	30	60	75	90	120	135	150	180	195	210	240
2	--	--	--	--	--	15	30	60	75	90	120	135	150	180	195	210	240
3	--	--	--	--	--	--	--	30	45	60	90	105	120	150	165	180	210
4	--	--	--	--	--	--	--	--	15	30	60	75	90	120	135	150	180
5	--	--	--	--	--	--	--	--	15	30	60	75	90	120	135	150	180
6	--	--	--	--	--	--	--	--	--	--	30	45	60	90	105	120	150
7	--	--	--	--	--	--	--	--	--	--	--	15	30	60	75	90	120
8	--	--	--	--	--	--	--	--	--	--	--	15	30	60	75	90	120
9	--	--	--	--	--	--	--	--	--	--	--	--	--	30	45	60	90
10	--	--	--	--	--	--	--	--	--	--	--	--	--	--	15	30	60
11	--	--	--	--	--	--	--	--	--	--	--	--	--	--	15	30	60
12	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	30
13	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
14	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
15	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
16	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Initial pheromone matrix: we assume the pheromone level in each arc is equal to 1.

τ (r,s)=

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	--	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	--	--	--	--	--	1	1	1	1	1	1	1	1	1	1	1	1
2	--	--	--	--	--	1	1	1	1	1	1	1	1	1	1	1	1
3	--	--	--	--	--	--	--	1	1	1	1	1	1	1	1	1	1
4	--	--	--	--	--	--	--	--	1	1	1	1	1	1	1	1	1
5	--	--	--	--	--	--	--	--	1	1	1	1	1	1	1	1	1
6	--	--	--	--	--	--	--	--	--	--	1	1	1	1	1	1	1
7	--	--	--	--	--	--	--	--	--	--	--	1	1	1	1	1	1
8	--	--	--	--	--	--	--	--	--	--	--	1	1	1	1	1	1
9	--	--	--	--	--	--	--	--	--	--	--	--	--	1	1	1	1
10	--	--	--	--	--	--	--	--	--	--	--	--	--	--	1	1	1
11	--	--	--	--	--	--	--	--	--	--	--	--	--	--	1	1	1
12	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	1
13	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
14	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
15	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
16	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

n=1

Ant 1

(1) In the initiation, the ant represents the vehicle 1, so $R_r^1 < R_m^1$. The ant is positioned on node 1 (R_r^1=2000+100=2100, $R_r^1 < R_m^1$). The set of reachable node is {5,6,7,8,9,10,11,12,13,14,15,16}. We generate q = 0.60 randomly. According to equation 3-2 the probability of reaching those nodes are 0.0666, 0.0333, 0.0166, 0.0133, 0.0111, 0.0083, 0.0074, 0.0066, 0.0055, 0.0051, 0.0048, and 0.0042. So the ant chooses node 5. Then we update the pheromone matrix according to equation 3-4($\tau(1,5)=(1-0.5)*1+0.5*1=1$). The ant visited node are {1,5}

τ (r,s)=

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	--	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	--	--	--	--	--	1	1	1	1	1	1	1	1	1	1	1	1
2	--	--	--	--	--	1	1	1	1	1	1	1	1	1	1	1	1
3	--	--	--	--	--	--	--	1	1	1	1	1	1	1	1	1	1
4	--	--	--	--	--	--	--	--	1	1	1	1	1	1	1	1	1
5	--	--	--	--	--	--	--	--	1	1	1	1	1	1	1	1	1
6	--	--	--	--	--	--	--	--	--	--	1	1	1	1	1	1	1
7	--	--	--	--	--	--	--	--	--	--	--	1	1	1	1	1	1
8	--	--	--	--	--	--	--	--	--	--	--	1	1	1	1	1	1
9	--	--	--	--	--	--	--	--	--	--	--	--	--	1	1	1	1
10	--	--	--	--	--	--	--	--	--	--	--	--	--	--	1	1	1
11	--	--	--	--	--	--	--	--	--	--	--	--	--	--	1	1	1
12	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	1
13	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
14	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
15	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
16	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

(2) At node 5(R_r^1=2100+25=2125, $R_r^1 < R_m^1$), the set of reachable node is {8,9,10,11,12,13,14,15,16}. We generate q = 0.65 randomly. According to equation 3-2 the probability of reaching those nodes are 0.0666, 0.0333, 0.0166, 0.0133, 0.0111, 0.0083, 0.0074, 0.0066 and 0.0055. So the ant chooses node 8. Then we update the pheromone matrix according to equation 3-4($\tau(5,8)=(1-0.5)*1+0.5*1=1$). The ant visited node are {1,5,8}.

(3) At node 8(R_r^1=2125+25=2150, $R_r^1 < R_m^1$), the set of reachable node is {11,12,13,14,15,16}. We generate q = 0.70 randomly. According to equation 3-2 the probability of reaching those nodes are 0.0666, 0.0333, 0.0166, 0.0133, 0.0111

and 0.0083. So the ant chooses node 11. Then we update the pheromone matrix according to equation 3-3($\tau(8,11)=(1-0.5)*1+0.5*1=1$). The ant visited node are {1,5,8,11}.

(4) At node 11(R_r^1=2150+25=2175, $R_r^1<R_m^1$), the set of reachable node is {14,15,16}. We generate q = 0.95 randomly. According to equation 3-3 the probability of reaching those nodes are 0.5714, 0.2857 and 0.1429. Then the ant chooses node 14. Then we update the pheromone matrix according to equation 3-4($\tau(11,14)=(1-0.5)*1+0.5*1=1$). The ant visited node are {1,5,8,11,14}.

(5) At node 14, the set of reachable node is { ϕ }. So the ant must return to start node to represent another vehicle.

(6) At start node, the ant represents vehicle 2, so R_r^2=2940 and $R_r^2<R_m^2$ The set of reachable node is {2,3,4,6,7,9,10,12,13,15,16}. According to equation 3-2, the ant choose node 2.

(7) At node 2(R_r^2=2940+25=2965, $R_r^2<R_m^2$), the set of reachable node is {6,7,9,10,12,13,15,16}. We generate q = 0.5 randomly. According to equation 3-2 the probability of reaching those nodes are 0.0333, 0.0166, 0.0111, 0.0083, 0.0074,0.0055, 0.0048 and 0.0042. So the ant chooses node 6. Then we update the pheromone matrix according to equation 3-4($\tau(2,6)=(1-0.5)*1+0.5*1=1$). The ant visited node are {1,2,5,6,8,11,14}.

(8) At node 6(R_r^2=2965+100=3065, $R_r^2>R_m^2$), so the ant must move to maintenance node. The set of reachable maintenance node is {M7,M8}. According to equation 3-2 the probability of reaching those maintenance nodes are 0.0333 and 0.0166.So the ant chooses maintenance node M7. Then we update the pheromone matrix according to equation 3-3($\tau(6,M7)=(1-0.5)*1+0.5*1=1$), and update the maintenance requirement travel mileage R_m^2=3000+3000=6000.

(9) At node M7, the set of reachable node is { ϕ }. So the ant must return to start node to represent another vehicle.

(10) At start node, the ant represents vehicle 3, so R_r^3=1000 and R_r^3<R_m^3 The set of reachable node is {3,4,7,9,10,12,13,15,16}. According to equation 3-2, the ant choose node 3.

(11) At node 3(R_r^3=1000+100=1100, R_r^3<R_m^3), the set of reachable node is {7,9,10,12,13,15,16}. We generate q = 0.55 randomly. According to equation 3-2 the probability of reaching those nodes are 0.0333, 0.0166, 0.0111, 0.0083, 0.0066,0.0055 and 0.0048. So the ant chooses node 7. Then we update the pheromone matrix according to equation 3-4($\tau(3,7)=(1-0.5)*1+0.5*1=1$). The ant visited node are {1,2,3,5,6,7,8,11,14}.

(12) At node 7(R_r^3=1100+100=1200, R_r^3<R_m^3), the set of reachable node is {12,13,15,16}. We generate q = 0.75 randomly. According to equation 3-2 the probability of reaching those nodes are 0.0333, 0.0166, 0.0111 and 0.0083. So the ant chooses node 12. Then we update the pheromone matrix according to equation 3-4(τ $(7,12)=(1-0.5)*1+0.5*1=1$). The ant visited node are {1,2,3,5,6,7,8,11,12,14}.

(13) At node 12(R_r^3=1200+100=1300, R_r^3<R_m^3), the set of reachable node is {16}. So the ant chooses node 16. Then we update the pheromone matrix according to equation 3-4(τ $(12,16)=(1-0.5)*1+0.5*1=1$). The ant visited node are {1,2,3,5,6,7,8,11,12,14,16}.

(14) At node 16, the set of reachable node is { ϕ }. So the ant must return to start node to represent another vehicle.

(15) At start node, the ant represents vehicle 4, so R_r^4=2500 and R_r^4<R_m^4 The set of reachable node is {4,9,10,13,15}. According to equation 3-2, the ant choose node 4.

(16) At node 4(R_r^4=2500+100=2600, $R_r^4<R_m^4$), the set of reachable node is {9,10,13,15}. We generate q = 0.80 randomly. According to equation 3-2, so the ant chooses node 9. Then we update the pheromone matrix according to equation 3-4($\tau(4,9)=(1-0.5)*1+0.5*1=1$). The ant visited node are {1,2,3,4,5,6,7,8,9,11,12,14,16}.

(17) At node 9(R_r^4=2600+100=2700, $R_r^4<R_m^4$), the set of reachable node is {13,15}. We generate q = 0.85 randomly. According to equation 3-2, so the ant chooses node 13. Then we update the pheromone matrix according to equation 3-4($\tau(9,13)=(1-0.5)*1+0.5*1=1$). The ant visited node are {1,2,3,4,5,6,7,8,9,11,12,13,14,16}.

(18) At node 13, the set of reachable node is { ϕ }. So the ant must return to start node to represent another vehicle.

(19) At start node, the ant represents vehicle 5, so R_r^5=1000 and $R_r^5<R_m^5$ The set of reachable node is {10,15}. According to equation 3-2, the ant choose node 10.

(20) At node 10(R_r^4=1000+100=1100, $R_r^5<R_m^5$), the set of reachable node is {15}. So the ant chooses node 15. Then we update the pheromone matrix according to equation 3-4($\tau(10,15)=(1-0.5)*1+0.5*1=1$). The ant visited node are {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16}.

(21) At node 7, the fit of reachable node is { ϕ }, and all nodes are visited by all ant. So the routes built by ant 1 are finished.

(22) The vehicle schedules of ant 1 are vehicle 1: {1,5,8,11,14}, vehicle 2: {2,6,M7}, vehicle 3: {3,7,12,16}, vehicle 4: {4,9,13}, vehicle 5: {10,15} The total cost is 5270(15+15+15+15+30+30+30+30+30+30+30+5*1000).

(23) We follow the same steps to find the solution of ant 2. The vehicle schedules of ant 2 are vehicle 1: {2,5,8,11,14}, vehicle 2: {1,M4,13}, vehicle 3: {3,7,12,16},

vehicle 4: {4,9,15}, vehicle 5:{6,10}. The total cost is 5270(15+15+15+15+30+30+30+30+60+30+5*1000).

(24) The routes built by ant 1 are the best. Then we update the pheromone matrix according to equation 3-5(ex. *τ(1,5)=(1-0.9)*1+0.9*1=1; τ(1,8)=(1-0.9)*1+0.5*0=0.1*).

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	0	--	1	1	1	1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	1	--	--	--	--	--	1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	2	--	--	--	--	--	0.1	1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	3	--	--	--	--	--	--	--	1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	4	--	--	--	--	--	--	--	--	0.1	1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	5	--	--	--	--	--	--	--	--	1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	6	--	--	--	--	--	--	--	--	--	--	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	7	--	--	--	--	--	--	--	--	--	--	--	0.1	1	0.1	0.1	0.1	0.1
τ (r,s)=	8	--	--	--	--	--	--	--	--	--	--	--	1	0.1	0.1	0.1	0.1	0.1
	9	--	--	--	--	--	--	--	--	--	--	--	--	--	1	0.1	0.1	0.1
	10	--	--	--	--	--	--	--	--	--	--	--	--	--	--	0.1	1	0.1
	11	--	--	--	--	--	--	--	--	--	--	--	--	--	--	1	0.1	0.1
	12	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	1
	13	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
	14	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
	15	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
	16	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Iteration 2

We follow the same steps to for iteration 2, and the results are listed below:

(1) The vehicle schedules of ant 1 are vehicle 1: {1,5,8,11,14}, vehicle 2: {2,6,M7}, vehicle 3: {3,7,12,16}, vehicle 4: {4,9,13}, vehicle 5: {10,15} The total cost is 5270(15+15+15+15+30+30+30+30+30+30+30+5*1000).

(2) . The vehicle schedules of ant 2 are vehicle 1: {2,6,10,15}, vehicle 2: {3,M5,14}, vehicle 3: {4,9,13}, vehicle 4: {1,5,8,11,16}, vehicle 5: {7,12} The total cost is 5285(30+30+30+30+30+15+15+15+60+30+5*1000).

(3) The best solution is ant 1.

We have reached the count of colonies (n = 2). So we stop the algorithm and output the result of this algorithm.

The result of this simple is shown by Table 4-2.

Table 4-2 The result of this simple

vehicle	schedule
1	1-5-8-11-14
2	2-6-M7
3	3-7-12-16
4	4-9-13
5	10-15

4.2 Real Example Test

In order to prove the algorithm of public transit scheduling with maintenance is useful, we obtain real operation data from a public transit company. These real operation data include vehicles, duties, maintenance requirement rules. In this section, the detail of the real example test is explained in the following subsection.

4.2.1 The Real Example Generation

We obtain real operation data for Tao-Yuan station from Tao-Yuan Bus Transportation Company on May 3, 2004. Tao-Yuan station has urban buses and inter-city buses. There are 320 duties and 55 vehicles operated. According to the schedule, 3 vehicles need to be maintained at that day. We also obtain maintenance requirement rules from this station.

4.2.2 Sensitive Analysis

We use 4 parts of these data to generate 4 small real examples to test the impacts of different parameters. In part 1, the network has 1 start node, 42 task nodes, 33 maintenance nodes (with 15 minutes interval from 8:00 to 17:00, maintenance requires 1 hour to finish.), and 1249 arcs (Each arc means a feasible connection of two nodes. It is considered a feasible connection if one node which the duty begin time late 10 minutes more than another node.). In part 2, the network has 1 start node, 53 task nodes, 33 maintenance nodes and 2462 arcs. In part 3, the network has 1 start node, 62 task nodes, 33 maintenance nodes and 3096 arcs. In part 4, the network has 1 start node, 66 task nodes, 33 maintenance nodes and 3314 arcs. In each experiment, we execute 100 iterations of one algorithm, assume the V_C is 1000 minutes.

In this sensitive analysis, we provide the improved percentage to represent the improvement of each test. The improve percentage is shown (4-1).

$$\textit{The Improved Percentage} = \frac{\textit{The Worst Case Object Value - The Best Case Object Value}}{\textit{The Worst Case Objective Value}} \times 100\% \quad (4\text{-}1)$$

There are six adjustable parameters compared as following:

1. The number of ants (n).
2. The initial pheromone deposited (initial $\tau(r,s)$).
3. The pheromone deposited of global updating rule ($\Delta\tau(r,s)$).
4. The pheromone deposited of local updating rule ($\Delta\tau_0$).
5. The pheromone decay parameter of global updating rule (α).
6. The pheromone decay parameter of local updating rule (ρ).

The detail of each test and result is explained as following:

1. The number of ants. (n)

In part 1, we test three cases to evaluate the impacts of number of ants. The number in case 1 is 5. The number in case 2 is 10. The number in case 3 is 15. The other parameters are given as following:

(1) The weight of pheromone trail is 2 (β = 2). The parameter is suggested by [4][11].

(2) The q_0 parameter is 0.9. The parameter is suggested by [4][11].

(3) The initial pheromone deposited is 0.01.

(4) The pheromone deposited of global updating rule is 0.0001.

(5) The pheromone deposited of local updating rule is $(10525*42)^{-1}$. The

10525 is a solution by greedy search. (This greedy search is we use our algorithm by n=1, m=1, q_0=1, β=1, α=1 and ρ=1 to solve this small real example.) The 42 is the number of task nodes.

(6) The pheromone decay parameter of global updating rule is 0.1. The parameter is suggested by [4][11].

(7) The pheromone decay parameter of local updating rule is 0.1. The parameter is suggested by [4][11].

The test results are shown in Table 4-3 and Figure 4-2.

Table 4-3 The test results of different number of ants (part1)

The number of ants (n)	Total operation idle time (Σt_l)	The number of vehicles used (V_{um})	The number of vehicles to maintenance	Objective value (Σt_l+ V_{um}*1000)
5	2295	8	1	10295
10	2315	8	1	10315
15	2315	8	1	10315

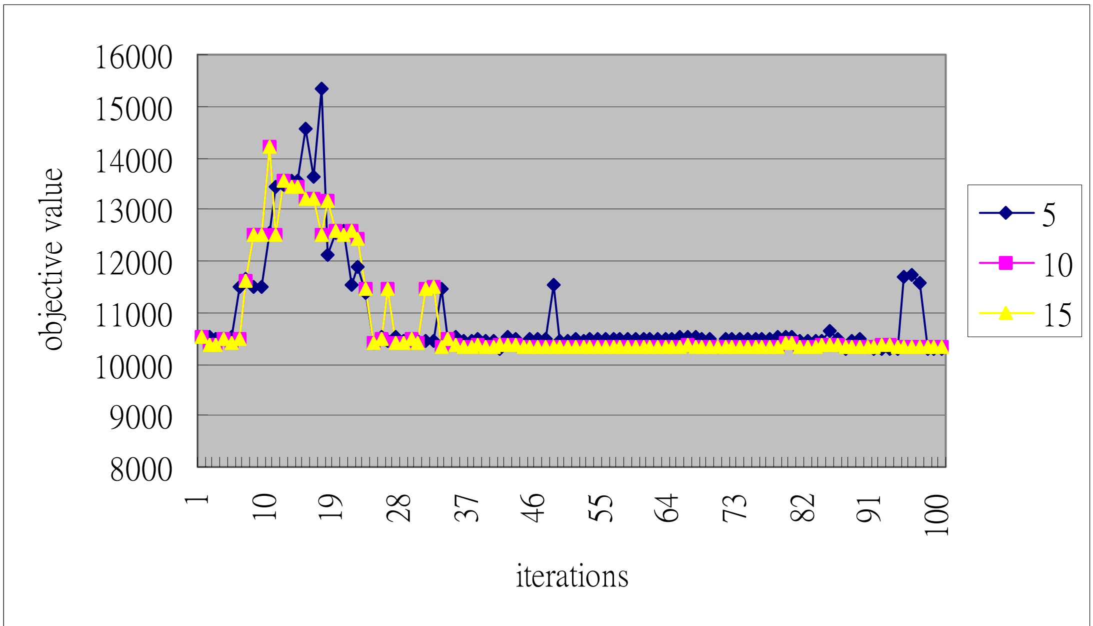

Figure 4-2 The test results of different number of ants (part1)

In this test, case 1 has the best objective value, whose ant number is 5. Both case 2 and case 3 has more number of ants, but the objective value is not better than case 1. So the more number of ants get good objective value is uncertainly. The improved percentage is 0.194%.

We follow the same steps to for other parts, and the results are shown in Table 4-4.

Table 4-4 The other parts results of different number of ants

Part 2			
Case	V_{um}	Objective value	Improved percentage
1	10	12682	0.45%
2	10	12740	
3	10	12740	
Part 3			
Case	V_{um}	Objective value	Improved percentage
1	11	14240	0.23%
2	11	14207	
3	11	14207	
Part 4			
Case	V_{um}	Objective value	Improved percentage
1	12	15238	9.45%
2	11	13798	
3	11	14110	

2. The initial pheromone deposited (initial $\tau(r,s)$).

We test three cases to evaluate the impacts of initial pheromone deposited. The initial pheromone deposited in case 1 is 1. The initial pheromone deposited in case 2 is 0.1. The initial pheromone deposited in case 3 is 0.01. The other parameters are given as following:

(1) The weight of pheromone trail is 2 (β = 2).

(2) The q_0 parameter is 0.9.

(3) The number of ants is 10. The parameter is suggested by [4][11].

(4) The pheromone deposited of global updating rule is 0.0001.

(5) The pheromone deposited of local updating rule is $(10525*42)^{-1}$.

(6) The pheromone decay parameter of global updating rule is 0.1.

(7) The pheromone decay parameter of local updating rule is 0.1.

The test results are shown in Table 4-5 and Figure 4-3.

Table 4-5 The test results of different initial pheromone deposited (part1)

The initial pheromone deposited	Total operation idle time (Σt_l)	The number of vehicles used (V_{um})	The number of vehicles to maintenance	Objective value ($\Sigma t_l + V_{um}*1000$)
1	2285	8	1	10285
0.1	2285	8	1	10285
0.01	2315	8	1	10315

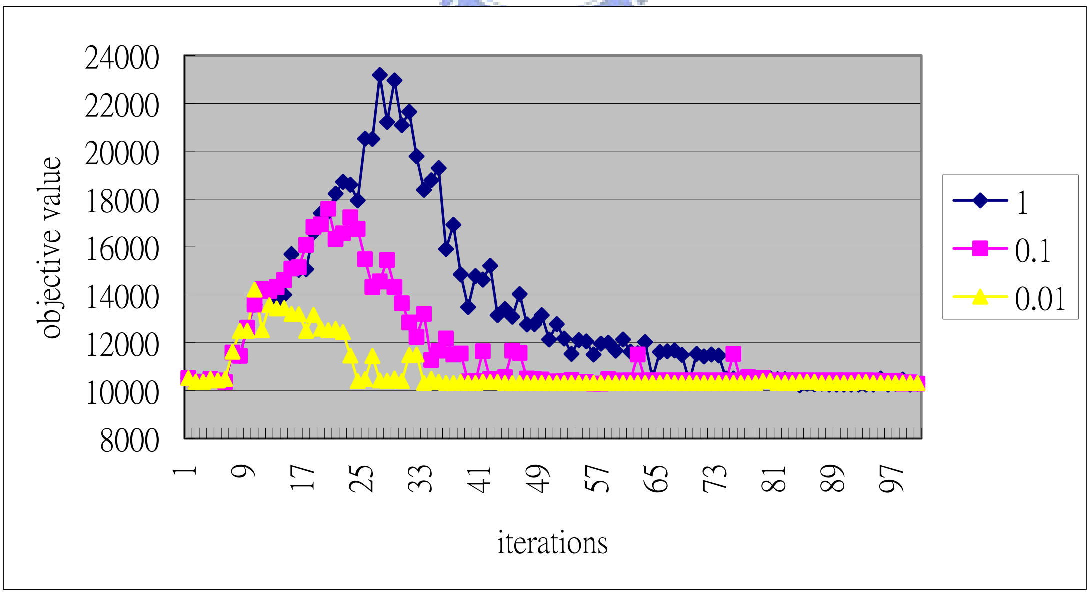

Figure 4-3 The test results of different initial pheromone deposited (part1)

In this test, when initial pheromone deposited is 1 or 0.1 can find better solutions. According to Figure 4-3, the case 1 needs more iterations than case 2 to converge, and the case 2 needs more iterations than case 3 to converge. So we can conclude the initial pheromone deposited is larger, the algorithm needs more iterations to converge, and the initial pheromone deposited is larger the impact of pheromone on arcs is larger. The improved percentage is 0.291%.

We follow the same steps to for other parts, and the results are shown in Table 4-6.

Table 4-6 The other parts results of different initial pheromone deposited

Part 2			
Case	V_{um}	Objective value	Improved percentage
1	10	12842	1.06%
2	10	12705	
3	10	12740	
Part 3			
Case	V_{um}	Objective value	Improved percentage
1	11	14234	0.62%
2	11	14296	
3	11	14207	
Part 4			
Case	V_{um}	Objective value	Improved percentage
1	12	15291	9.75%
2	11	14079	
3	11	13798	

3. The pheromone deposited of global updating rule ($\Delta\tau(r,s)$).

We test three cases to evaluate the impacts of pheromone deposited of global updating rule. The pheromone deposited of global updating rule in case 1 is 0.01. The pheromone deposited of global updating rule in case 2 is 0.001. The pheromone deposited of global updating rule in case 3 is 0.0001. The other parameters are given as following:

(1) The weight of pheromone trail is 2 ($\beta = 2$).

(2) The q_0 parameter is 0.9.

(3) The number of ants is 10.

(4) The initial pheromone deposited is 1.

(5) The pheromone deposited of local updating rule is $(10525*42)^{-1}$.

(6) The pheromone decay parameter of global updating rule is 0.1.

(7) The pheromone decay parameter of local updating rule is 0.1.

The test results are shown in Table 4-7 and Figure 4-4.

Table 4-7 The results of different pheromone deposited of global updating rule (part1)

The pheromone deposited of global updating rule ($\Delta \tau(r,s)$)	Total operation idle time (Σt_l)	The number of vehicles used (V_{um})	The number of vehicles to maintenance	Objective value ($\Sigma t_l + V_{um}*1000$)
0.01	2295	8	1	10295
0.001	2294	8	1	10295
0.0001	2285	8	1	10285

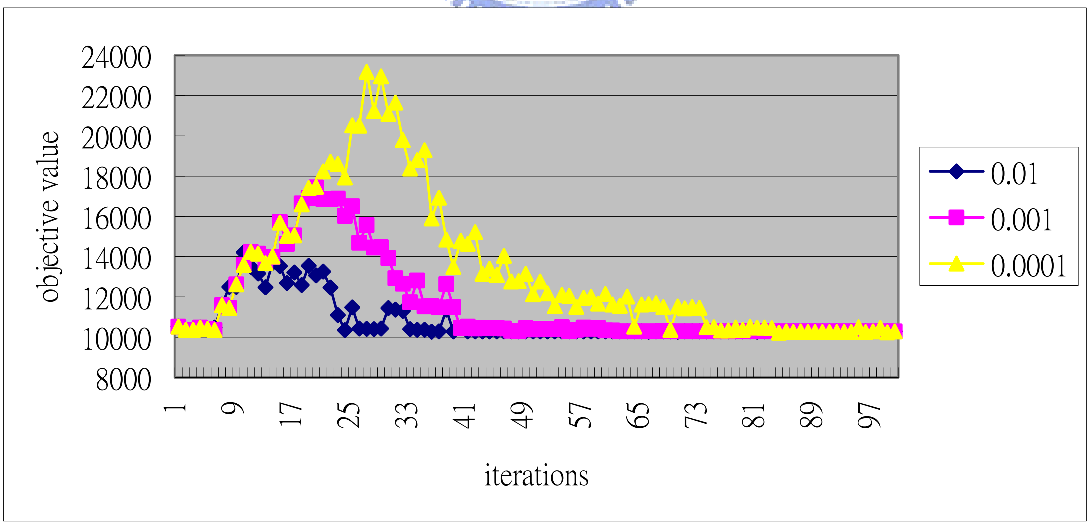

Figure 4-4 The results of different pheromone deposited of global updating rule (part1)

In this test, the best solution is found when the pheromone deposited of global updating rule is 0.0001. According to Figure 4-4, the case 3 needs more iterations than case 2 to converge, and the case 1 needs more iterations than case 2 to converge. So we can conclude the pheromone deposited of global updating rule is larger, the algorithm needs less iterations to converge, and the pheromone deposited of global updating rule is larger the impact of pheromone on arcs is larger. The improved percentage is 0.097%.

We follow the same steps to for other parts, and the results are shown in Table 4-8.

Table 4-8 The other parts results of different deposited of global updating rule

Part 2			
Case	V_{um}	Objective value	Improved percentage
1	10	12657	1.45%
2	10	12655	
3	10	12482	
Part 3			
Case	V_{um}	Objective value	Improved percentage
1	11	14297	0.44%
2	11	14260	
3	11	14234	
Part 4			
Case	V_{um}	Objective value	Improved percentage
1	11	14120	9.36%
2	11	13859	
3	12	15291	

4. The pheromone deposited of local updating rule ($\Delta\tau_0$).

We test three cases to evaluate the impacts of pheromone deposited of local updating rule. The pheromone deposited of local updating rule in case 1 is 0.00001. The pheromone deposited of local updating rule in case 2 is 0.000001. The pheromone deposited of local updating rule in case 3 is 0.0000001. The other parameters are given as following:

(1) The weight of pheromone trail is 2 (β = 2).

(2) The q_0 parameter is 0.9.

(3) The number of ants is 10.

(4) The initial pheromone deposited is 0.01.

(5) The pheromone deposited of global updating rule is 0.0001.

(6) The pheromone decay parameter of global updating rule is 0.1.

(7) The pheromone decay parameter of local updating rule is 0.1.

The test results are shown in Table 4-9 and Figure 4-5.

Table 4-9 The results of different pheromone deposited of local updating rule (part1)

The pheromone deposited of local updating rule ($\Delta\ \tau_0$)	Total operation idle time (Σt_l)	The number of vehicles used (V_{um})	The number of vehicles to maintenance	Objective value ($\Sigma t_l + V_{um}*1000$)
0.00001	2245	8	1	10245
0.000001	2245	8	1	10245
0.0000001	2182	8	1	10182

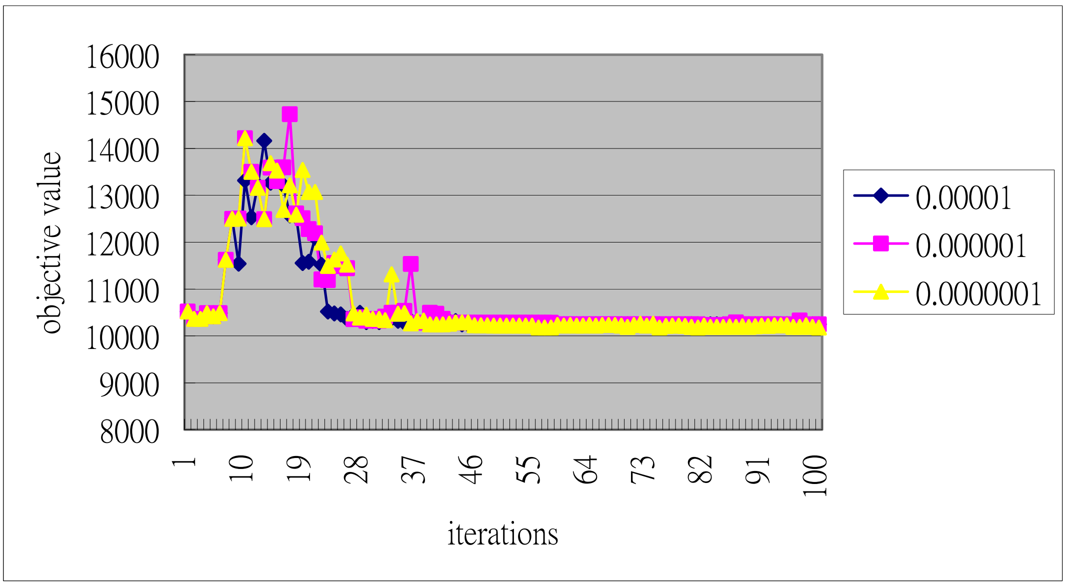

Figure 4-5 The results of different pheromone deposited of local updating rule (part1)

In this test, the best solution is found when pheromone deposited of local updating rule is 0.0000001. According to Figure 4-5, we can conclude the pheromone deposited of local updating rule is not sensitive on the number of iterations of the solution to converge. The improved percentage is 0.615%.

We follow the same steps to for other parts, and the results are shown in Table 4-10.

Table 4-10 The other parts results of different deposited of local updating rule

Part 2			
Case	V_{um}	Objective value	Improved percentage
1	11	13725	7.72%
2	10	12722	
3	10	12665	
Part 3			
Case	V_{um}	Objective value	Improved percentage
1	11	14203	0.35%
2	11	14253	
3	11	14208	
Part 4			
Case	V_{um}	Objective value	Improved percentage
1	11	14238	3.84%
2	11	13691	
3	11	13976	

5. The pheromone decay parameter of global updating rule (α).

We test three cases to evaluate the impacts of pheromone decay parameter of global updating rule. The pheromone decay parameter in case 1 is 0.1. The pheromone decay parameter in case 2 is 0.5. The pheromone decay parameter in case 3 is 0.9. The other parameters are given as following:

(1) The weight of pheromone trail is 2 (β = 2).

(2) The q_0 parameter is 0.9.

(3) The number of ants is 10.

(4) The initial pheromone deposited is 0.01.

(5) The pheromone deposited of global updating rule is 0.0001.

(6) The pheromone deposited of local updating rule is $(10525*42)^{-1}$.

(7) The pheromone decay parameter of local updating rule is 0.1.

The test results are shown in Table 4-11 and Figure 4-6.

Table 4-11 The results of different decay parameter of global updating rule (part1)

The pheromone decay parameter of global updating rule (α)	Total operation idle time (Σt_l)	The number of vehicles used (V_{um})	The number of vehicles to maintenance	Objective value ($\Sigma t_l + V_{um}*1000$)
0.1	2315	8	1	10315
0.5	2335	8	1	10335
0.9	2365	8	1	10365

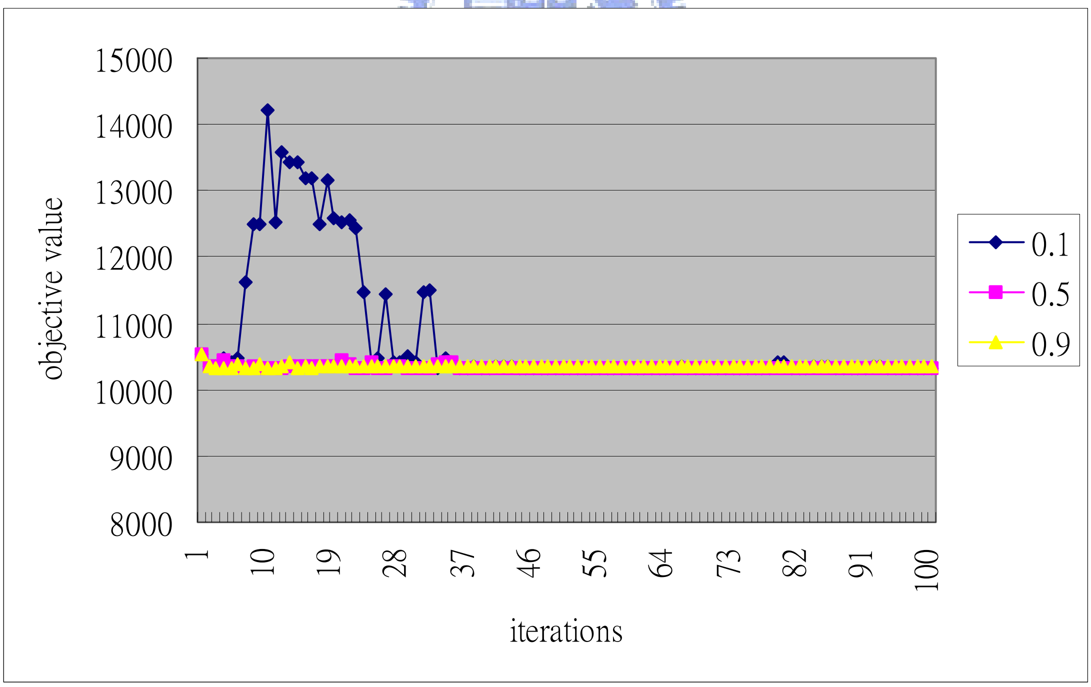

Figure 4-6 The results of different decay parameter of global updating rule (part1)

In this test, the best solution is found when pheromone decay parameter of global updating rule is 0.1. According to Figure 4-6, the case 1 needs same iterations to converge, and we can conclude the pheromone decay parameter of global updating rule is larger the solution is sensitive impact on the late solution and the pheromone decay parameter of global updating rule is larger the impact of pheromone on arcs is larger. The improved percentage is 0.482%.

We follow the same steps to for other parts, and the results are shown in Table 4-12.

Table 4-12 The other parts results of different decay parameter of global updating rule

Part 2			
Case	V_{um}	Objective value	Improved percentage
1	10	12740	0.35%
2	10	12720	
3	10	12765	
Part 3			
Case	V_{um}	Objective value	Improved percentage
1	11	14207	0.39%
2	11	14263	
3	11	14239	
Part 4			
Case	V_{um}	Objective value	Improved percentage
1	11	13798	9.23%
2	11	14241	
3	12	15202	

6. The pheromone decay parameter of local updating rule (ρ).

We test three cases to evaluate the impacts of pheromone decay parameter of global updating rule. The pheromone decay parameter in case 1 is 0.1. The pheromone decay parameter in case 2 is 0.5. The pheromone decay parameter in case 3 is 0.9. The other parameters are given as following:

(1) The weight of pheromone trail is 2 (β = 2).

(2) The q_0 parameter is 0.9.

(3) The number of ants is 10.

(4) The initial pheromone deposited is 0.01.

(5) The pheromone deposited of global updating rule is 0.0001.

(6) The pheromone deposited of local updating rule is $(10525*42)^{-1}$.

(7) The pheromone decay parameter of global updating rule is 0.1.

The test results are shown in Table 4-13 and Figure 4-7.

Table 4-13 The results of different decay parameter of local updating rule (part1)

The pheromone decay parameter of local updating rule (ρ)	Total operation idle time (Σt_l)	The number of vehicles used (V_{um})	The number of vehicles to maintenance	Objective value ($\Sigma t_l + V_{um}*1000$)
0.1	2315	8	1	10315
0.5	2270	8	1	10270
0.9	2285	8	1	10285

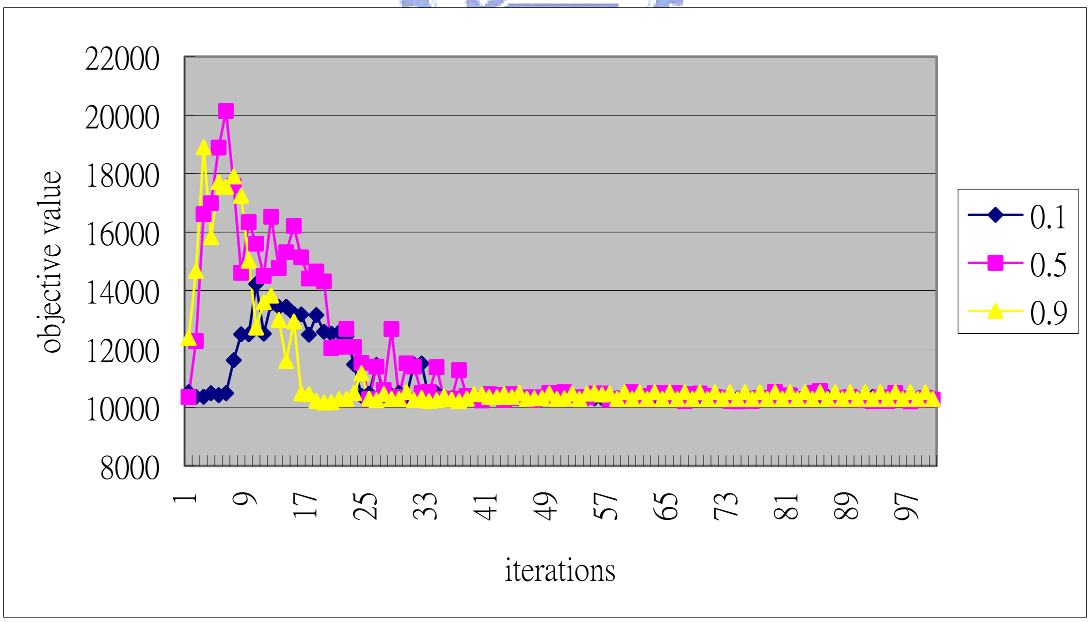

Figure 4-7 The results of different decay parameter of local updating rule (part1)

In this test, the best solution is found when pheromone decay parameter of local updating rule is 0.5. According to Figure 4-7, we can infer the pheromone decay parameter of local updating rule is larger, the algorithm needs less iterations to converge. The improved percentage is 0.436%.

We follow the same steps to for other parts, and the results are shown in Table 4-14.

Table 4-14 The other parts results of different decay parameter of local updating rule

Part 2			
Case	V_{um}	Objective value	Improved percentage
1	10	12740	2.35%
2	10	12535	
3	10	12440	
Part 3			
Case	V_{um}	Objective value	Improved percentage
1	11	14207	0.39%
2	11	14228	
3	11	14263	
Part 4			
Case	V_{um}	Objective value	Improved percentage
1	11	13798	1.60%
2	11	13650	
3	11	13873	

4.3 Summary

The part 1 in the real operation used 8 vehicles, and the objective value is 10700. The part 2 in the real operation used 10 vehicles, and the objective value is 13047. The part 3 in the real operation used 11 vehicles, and the objective value is 14233. The part 4 in the real operation used 11 vehicles, and the objective value is 14634. The test and analysis is summarized as following:

1. The algorithm can successfully solve the problem of real operation, and generate the buses operation schedule and maintenance schedule at the same time.
2. The algorithm can make the buses schedule have better objective value than

the real operation.

3. According to this sensitive analysis, the percentage of improve is bed in each test. So we can conclude the parameters setting are not sensitive on each test in this situation, we execute 100 iterations of our algorithm.

Chapter5 Conclusions and Suggestions

5.1 Conclusions

For the moment, the achievements progress of this research is shown as following:

1. Develop the algorithm for bus operation scheduling which takes vehicle maintenance requirement into consideration: We propose an ant colony system (ACS) based algorithm to solve the bus scheduling with maintenance requirement.
2. According to the algorithm, we can make the buses operation and maintenance schedules work at the same time.
3. The idle time of the buses operation schedule made by our algorithm is less than the real operation schedule idle time.
4. According to the algorithm, we can decrease the number of used vehicles. So the company can save the cost of vehicle purchasing.

5.2 Suggestions

The topics we suggest for future research are:

1. Considering working habits of drivers: In this research, we don't consider the driver’s working time and behavior. We suggest considering the driver's working habits in comparison to artificial ants.
2. Consider the characteristics of routes: In this research, we don’t consider the characteristics of routes. In a real operation, the scheduling of vehicles is usually concerned about the characteristics of routes. We suggest considering the characteristics of routes with the algorithm.
3. Modify the solutions of last several vehicles: The idle time of last several

vehicles solved by our algorithm is a lot. We suggest modifying the solution of last several vehicles.

4. Consider the maintenance capacity: In this research, we don't consider the maintenance capacity. We suggest considering the maintenance capacity in maintenance nodes.
5. Improve the pheromone updating rules: In our algorithm, the pheromone updating rules are followed by original rules. We suggest improving the pheromone updating rules to find a better solution.

References

[1]. Ali Haghiani, Yousef Shafahi, “Bus maintenance systems and maintenance scheduling: model formulations and solutions”, *Transportation Research Part A*, 36, pp.453-482, 2002.

[2]. Walid El Moudani, Félix Mora-Camino, “A dynamic approach for aircraft assignment and maintenance scheduling by airlines”, *Journal of Air Transport Management*, 6, pp.233-237, 2000.

[3]. Ta-Hui Yang, Sangyao Yan, Hsuan-Huan Chen, “An airline maintenance manpower planning model with flexible strategies”, Journal of Air Transport Management, 9, pp.233-239, 2003.

[4]. Marco Dorigo, Luca Maria Gambardella, “Ant colony system: A cooperative learning approach to traveling salesman problem”, *IEEE Transactions on Evolutation*, Vol. 1, No.1, pp.53-66, 1997.

[5]. Marco Dorigo, Gianni Di Caro, Luca M. Gambardella, ”Ant algorithm for discrete Optimization”, Artificial Life, Vol.5, No.3, pp.137-172, 1999.

[6]. Bernd Bullnhetmer, Richard F. Hartl, Christine Strauss, “Applying the ant System to the vehicle routing problem”, 2nd International Conference on Metaheuristics – MIC97, Sophia-Antipolis, France, July 21-24, 1997.

[7]. Bernd Bullnhetmer, Richard F. Hartl, Christine Strauss, “An improved ant system for the Vehicle Routing Problem”, Preprint 1, pp.1-11, 1997.

[8]. Min-Hua Lo, “A new ant colony algorithm to vehicle routing problem under capacity and distance constraints”, master thesis, Yuan-Ze University, 2003.

[9]. Paul Forsyth, Anthony Wren, “An ant system for bus driver scheduling”, 7^{th} International Workshop on Computer-Aided Scheduling of Public Transport, Boston, July 1997.

[10]. Marco Dorigo, Gianni Di Caro, “Ant colony optimization: A new meta-huristic”, IEEE 1999.

[11]. Marco Dorigo, Luca Maria Gambardella, “Ant colonies for traveling salesman problem”, Accepted for publication in BioSystems, 1997.