Journal of the Operations Research Society of China
https://doi.org/10.1007/s40305-020-00307-8

# Optimization of Correspondence Times in Bus Network Zones, Modeling and Resolution by the Multi-agent Approach

**Elbaz Hassane**[1] **· Elhilali Alaoui Ahmed**[2]

Received: 21 July 2019 / Revised: 18 January 2020 / Accepted: 15 April 2020


## Abstract

Urban transportation, especially bus transportation, is an important sign of development in every city in the world. The average waiting time for passengers at correspondence stations of buses is one of the most important measures of effectiveness of bus transportation. To the best of our knowledge, the studies in the literature are about maximizing the number of synchronizations in those correspondence stations whose objective is to minimize the waiting time in the network. The classical definition of synchronization used in the literature related to a time window. In this work, we introduce a new definition of synchronization of two buses in network zones. Within this context, we present a mathematical formulation of the synchronization bus timetabling problem as a multi-objective program, where we use the new meaning for synchronization of two buses in the network zones. Since the problem is NP-hard, we adapt a multi-agent approach to solve it. Numerical experiments show that after adapting the multi-agent approach using our proposed definition, we obtain high-quality solutions compared to the classical definition.

**Keywords** Multi-objective problem · Bus transportation · Network zones · Multi-agent approach · Synchronization

**Mathematics Subject Classification** 90B20

✉ Elbaz Hassane
hassane.elbaz@usmba.ac.ma

Elhilali Alaoui Ahmed
a.elhilali-alaoui@ueuromed.org

1 Modeling and Scientific Computing Laboratory, Faculty of Science and Technology, Fez, Morocco

2 Euromed University, Fez, Morocco

Published online: 10 July 2020

## 1 Introduction

Nowadays, public transport systems are widely identified as a potential way to reduce pollution, to reduce energy consumption, to improve mobility, to reduce traffic congestion and to create new jobs. For this reason, several works have been developed in this area, especially in the bus transportation area, which is the backbone of the public transport network. Bus network planning is distributed in several main areas such as [1]:

– designing the network lines,
– scheduling the bus network,
– bus planning, and
– crew planning.

The second area is one of the most important tasks. Indeed, waiting time is perceived in a very negative way by users who associate it with uncontrollable wasted time. For a good mobility of passengers in the urban bus transport network, it is necessary to ensure possibilities of correspondence between the different lines. The quality of correspondence depends on the waiting time of passengers in the transfer nodes.

The problem is to schedule the buses in an urban transport network in order to improve the quality of service by minimizing the waiting time of passengers at the different stations, thus creating a maximum synchronization that allows the transfer of passengers from one line to another with the minimum waiting time in the correspondence zones. We begin with giving the definition of synchronization used in the literature, and after we present a new definition of the synchronization between two buses in a corespondance zone on which we base our modeling. Then, in two examples, we show the advantages of our definition in relation to the definition existing in the literature. We apply it to propose a mathematical model. Thereafter, we propose a resolution method based on the multi-agent approach composed of several agents which concurrently explore the search space and cooperate to improve and to coordinate their behaviors. Our goal is to determine the departure time of every bus trip in the network, without changing the frequencies. This process can be done either by changing the service frequencies or keeping the pre-determined frequencies for the lines [2]. In the first case, it can result in changing the required fleet size (e.g., adding more buses to a bus route). The second case does not affect the fleet size and enables planners to reduce transfer waiting time by existing resources.

## 2 State of the Art

The planning of bus timetabling is one of the important tasks in the exploitation and scheduling of public transport. It has been addressed in several studies recently where the objective is to improve the quality of service offered to passengers. Among the first ideas is to minimize the total transfer waiting time, based on the idea of preventing extremely long waiting times for transferring passengers, which surely discourage transit users [3]. A mathematical model of the problem was proposed in Ceder et al. [4], where they minimized the transfer time at the correspondence stations by maximizing the number of buses arriving at the same station simultaneously. The problem is solved for a small transit network by a heuristic on instances involving 14 lines. An extension of this work presented by Eranki [5], which linked the synchronization to a time window, that is, maximize the number of buses arriving at the stations in a time window by heuristics on instances with a maximum 6 lines and 9 nodes maximum [5]. Other research attempted to, like [6,7], emphasize that this problem is an extremely difficult case even for small transit networks. For this reason, they have treated small instances. Zhigang et al. [8] reformulated the timetabling problem presented by [4] and implemented the Nesting Taboo Search (NTS) to solve the problem [8]. Jansen et al. [1] proposed a method to synchronize bus timetables in order to minimize the transfer time of passengers in the network with fixed headways between the buses of the same line. The tab search was applied to solve the problem. The model was tested on the city of Copenhagen [1]. Ibarra-Rojas and Rios-Solis [9] proved that the complexity of the bus timetabling problem is NP-hard. To solve this problem, they use a multi-start iterated local search algorithm to maximize the number of synchronizations in the network [9]. Fouilhoux et al. [10] defined four families of valid inequalities to solve the problem of synchronization with an exact method but for small and medium instances. Feng et al. [11] developed, by optimizing the layout of the transit routes, transit network optimization model that considers reducing not only time-consuming transfers but also trips which have to make relatively more transfers. Ibarra-Rojas et al. [12] presented an integrated optimization problem for frequency setting, timetabling, and passenger assignment to minimize the sum of passenger and operation costs.

In this study, we redefine the synchronization between buses in the network. To the best of our knowledge, this is the only definition used in the literature related to a time window, where we find a lower bound and a higher bound. We propose a multi-objective optimization problem based on the new definition to model the problem. In the resolution step, an exact method is not practicable. Then, we adapted the multi-agent approach to solve it.

## 3 Problem Description

According to Guihaire [13], among the general criteria for measuring the quality of service in a public transport network, we find for example:

- Easy mobility in the network (high number of possible connections, short waiting times).
- Minimized total travel time (synchronized connections, high travel speeds).
- Good accessibility of the network.

In this paper, we focused on the total waiting time in the transfer stations. Generally, the bus network planning process consists of configuration of the network lines, timetable planning, vehicle planning and crew planning [10].

In this paper, the problem treated is the timetable generation which represents one of the best criteria to measure the quality of service. The aim is to improve bus schedules in urban areas to meet the demand of passengers by maximizing quality of service. We focus on the synchronization bus timetabling problem where the objective is to find a solution minimizing the waiting time for every passenger, especially in the correspondence zones. Before attacking that, we consider the travel time to be composed of the following phases: the time necessary to arrive at the first stop of the network, the travel time on the bus, the waiting time in the first stop of the network, the transfer time from one bus to another...

From the passenger's point of view, these types of waiting times do not have the same importance. Several studies have shown that off-vehicle waiting times are more annoying to passengers compared to the time spent in the vehicle [14–16]. Specifically, they have shown that the waiting time in the correspondence stations is more important than others [15], because on the one hand, if the departure time is fixed, the passenger can arrive at a departure station without an initial waiting time. On the other hand, the waiting time in the correspondence stations is fixed according to the departure times of buses.

Figure 1 illustrates an example of a transfer zone. In the zoom of the transfer zone, we find that, in order to change buses from the line $i$ to the line $j$ in the transfer zone, the passenger needs a walking time.

To increase the number of synchronizations in a network of buses with minimal time implies an improvement of the quality of service. For this, our contribution is to maximize the number of buses synchronized and, at the same time, to minimize the waiting time of passengers between synchronized buses. In order to minimize the waiting time of a passenger transferring from line $i$ to line $j$, several studies have treated this problem by maximizing the number of synchronized buses using the synchronization clarified in Fig. 2.

In this figure, there is green time window and only buses in this window are synchronized (the bars are the arrival times at a bus stop), meanings that the bus $p$ synchronizes

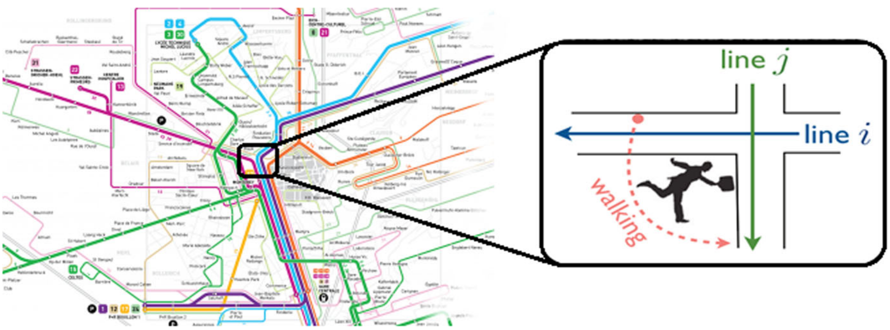



**Fig. 1** Zoom in of transfer zone

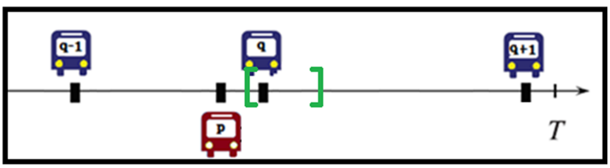



**Fig. 2** Example of synchronization

with the bus $q$ in the same station (the bus $q$ belongs to the time window associated to the bus $p$).

In this paper, we propose the following definition: We say that a bus $p$ from line $i$ synchronizes with a bus $q$ from another line $j$ if the arrival time of trip $(q, j)$ arrives at correspondence station of after the arrival time of trip $(p, i)$ and verified the following assumptions:

- The bus $q$ arrives after the arrival time of bus $p$ + the walking time.
- There is no bus arriving between $p$ and $q$ that verifies the previous assumption.

Maximizing the number of buses synchronized using the classical definition does not minimize the waiting time in a practical way. To show the advantages of our definition, we study the two following examples. In the first example, we take the cases shown in Fig. 3. In both cases, the bus $p$ synchronizes with the bus $q$ even though the waiting time differs from one case to another. So, if we maximize the number of synchronized buses, there is always a waiting time that is not necessary to have a synchronization, so in our modeling we always take the case that minimizes the waiting time in an effective way.

Another example is illustrated in Fig. 4. The bus $p$ does not synchronize with $q$, so we cannot reduce the time gap between the arrivals of the bus $p$ and the bus $q$ even if there is a possibility.

## 4 Problem Formulation

The objective functions of the model are the maximization of the synchronization buses using the new definition and the minimization of the total waiting time in the

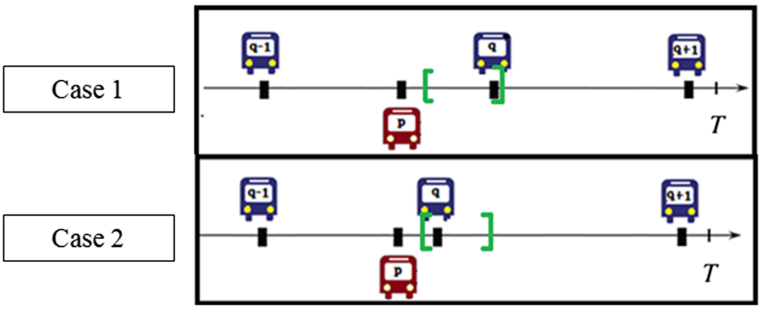



**Fig. 3** Example of two synchronizations

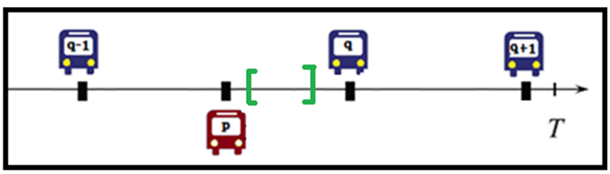



**Fig. 4** Example of two unsynchronized buses

transfer zones. The output of the model is the departure time of buses from the first station of each line. In this section, we present a mathematical formulation of the synchronization bus timetabling problem.

### 4.1 Notations

**Sets and parametrs:**

- $I$: set of lines in the bus network;
- $I(i)$: set of lines that may have a synchronization nodes with line $i$;
- $S(i, j)$: represents all the synchronization nodes for pair of lines $(i,j)$;
- $F_i$: number of buses in service on route $i$ (number of trips);
- $N_s^{ij}$: average of passengers transferring between two buses, from line $i$ to line $j$ in correspondence node $s$;
- $h^i_{\min}$: minimum headway (minimum time between each successive departures) of the line $i$;
- $h^i_{\max}$: maximum headway (maximum time bet;ween each successive departures) of the line $i$;
- $t_i^s$: arrival time at the transfer zone $s$ by the line $i$;
- $\omega_{ijs}$: walking time to move from line $i$ to line $j$ in the zone $s$ due to the geometric form of the station;
- $\Delta_{sip}^{jq}$: duration between the arrival of the $p$th bus of line $i$ and the $q$th bus of line $j$ in the station $s$.

Buses generally do not arrive exactly according to their scheduled time. It seems appropriate to use a probability distribution that reflects the differences of scheduled times from real-time bus [6].

- $\Delta_{isp}$: deviation of scheduled times due to random bus arrivals

**Decision variables**

- $Y_{sip}^{jq} = \begin{cases} 1, & \text{if the } p\text{th trip of line } i \text{ synchronizes with the } q\text{th trip of line } j \text{ in zone } s, \\ 0, & \text{otherwise.} \end{cases}$
- $X_i^p$: represent the departure time of the $p$th bus in line $i$

### 4.2 Multi-objective Model

The mathematical model that we propose in this work contains two objectives where the global aim is to increase the quality of service by minimizing waiting time in the correspondence zones and to maximize the operational benefits of the bus company by creating new lines with correspondences in the network.

- The first objective is to maximize the number of buses synchronized.

$$\text{Max } Z = \sum_{i\in I} \sum_{j\in I(i)} \sum_{s\in S(i,j)} \sum_{p=1}^{F_i} \sum_{q=1}^{F_j} N_s^{ij} Y_{sip}^{jq}.$$

- The second objective is to minimize the waiting time in the synchronization stations.

$$\text{Min } Z = \sum_{i \in I} \sum_{j \in I(i)} \sum_{s \in S(i,j)} \sum_{p=1}^{F_i} \sum_{q=1}^{F_j} N_s^{ij} Y_{sip}^{jq} \Delta_{sip}^{jq}.$$

- The constraints of the model are as follows:

$$\Delta_{sip}^{jq} = (X_j^q + t_j^s + \Delta_{jsq}) - (X_i^p + t_i^s + \Delta_{isp} + \omega_{ijs}), \quad \forall i \in I, \forall j \in I(i), \forall s \in S(i,j), \quad \forall p \in \{1 \cdots F_i\}, \forall q \in \{1 \cdots F_j\}; \tag{4.1}$$

$$X_i^{F_i} \leqslant T, \quad \forall i \in I; \tag{4.2}$$

$$h_{\min}^i \leqslant X_i^{p+1} - X_i^P, \quad \forall i \in I, \forall P \in \{1 \cdots F_i - 1\}; \tag{4.3}$$

$$X_i^{p+1} - X_i^P \leqslant h_{\max}^i, \quad \forall i \in I, \forall P \in \{1 \cdots F_i - 1\}; \tag{4.4}$$

$$M(Y_{sip}^{jq} - 1) \leqslant \Delta_{sip}^{jq}, \quad \forall i \in I, \forall j \in I(i), \forall s \in S(i,j), \quad \forall p \in \{1 \cdots F_i\}, \forall q \in \{1 \cdots F_j\}; \tag{4.5}$$

$$\Delta_{sip}^{jq} Y_{sip}^{j(q+1)} \leqslant 0, \quad \forall i \in I, \forall j \in I(i), \forall s \in S(i,j), \quad \forall p \in \{1 \cdots F_i\}, \forall q \in \{1 \cdots F_j - 1\}; \tag{4.6}$$

$$\sum_{k=1}^{F_j} Y_{sip}^{jk} \leqslant 1, \quad \forall i \in I, \forall j \in I(i), \forall s \in S(i,j), \quad \forall p \in \{1 \cdots F_i\}; \tag{4.7}$$

$$\Delta_{sip}^{jq} Y_{sip}^{jq} \leqslant X_i^{p+1} - X_i^P, \quad \forall i \in I, \forall j \in I(i), \forall s \in S(i,j), \quad \forall p \in \{1 \cdots F_i - 1\}, \forall q \in \{1 \cdots F_j\}; \tag{4.8}$$

$$Y_{sip}^{jq} \in \{0, 1\}, \quad \forall i \in I, \forall j \in I(i), \forall s \in S(i,j), \quad \forall p \in \{1 \cdots F_i\}, \forall q \in \{1 \cdots F_j\}; \tag{4.9}$$

$$X_i^P \in \mathbb{N}^+, \quad \forall i \in I, \forall p \in \{1 \cdots F_i\}. \tag{4.10}$$

The first constraint (4.1) defines $\Delta_{sip}^{jq}$ as the necessary waiting time of a passenger to transfer from the bus $q$ to the bus $p$ in the zone $s$. The second constraint (4.2) ensures that the departures of buses in each line belong to the planning horizon $T$. The constraint (4.3) (resp. (4.4)) indicates the minimum (resp. maximum) time limit between two adjacent buses of the same line. The constraint (4.5) ensures that a bus of the line $i$ does not synchronize with the buses already passed in the line $j$ at the synchronization station that is to say if $\Delta_{sip}^{jq} < 0$ then $Y_{sip}^{jq} = 0$. The sixth and seventh constraints (4.6) and (4.7) ensure that each bus $p$ of the line $i$ synchronizes at most with one bus (of the line $j$) which arrives directly after the bus $p$. We assume that the bus $q$ can take all passengers who are on bus $p$ of line $i$ and who wish to change line $i$ to line $j$ in station $s$ without waiting for bus $q + 1$ of line $j$. In the case where

$\Delta_{sip}^{jq} > 0$, there will be no synchronization between the bus $p$ of line $i$ and the bus $p+1$ of line $j$. Indeed, necessarily we have $Y_{sip}^{j(q+1)} = 0$, illustrated in Fig. 5.

The constraint (4.8) ensures that we cannot have two buses of the line $i$ synchronizing with the same bus of the line $j$. To clarify this, let us take the following example (Fig. 6).

In this case, we have that $\Delta_{sip}^{jq} < X_i^{p+1} - X_i^p$ implies $Y_{sip}^{j(q)} = 0$. Therefore, there will be no synchronization between bus $p$ of line $i$ and bus $q$ of line $j$, but there will be synchronization between bus $p+1$ of line $i$ and bus $q$ of line $j$. So the passengers who want to change buses from line $i$ to line $j$ in station $s$ must choose the bus $p+1$ at the start of the line to avoid more waiting time to change buses in station $s$. The last two constraints (4.9) and (4.10) define the decision domain of variables.

Before solving the problem, we must check the following two inequalities. The problem is impractical unless these two inequalities are true.

$$T \geqslant (F_i - 1)h_{\min}^i,$$
$$h_{\max}^i \geqslant h_{\min}^i.$$

The first inequality ensures that all bus departures belong to the planning horizon, because if the inequality is not verified, then the last departure from line $i$ does not belong to the planning horizon $T$. The second inequality verifies whether the minimum headway is less than the maximum headway of each line; otherwise, the problem is impracticable [4].

## 5 Resolution Approach

The use of exact methods to find the optimal solution requires a fairly long computation time that grows exponentially depending on the size of the problem addressed.

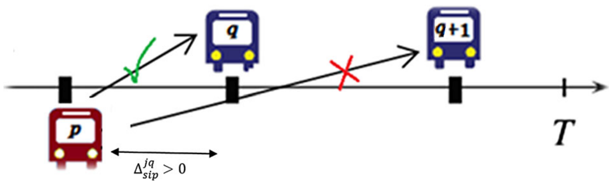



**Fig. 5** Example of a non-synchronization between the bus $p$ and $q+1$

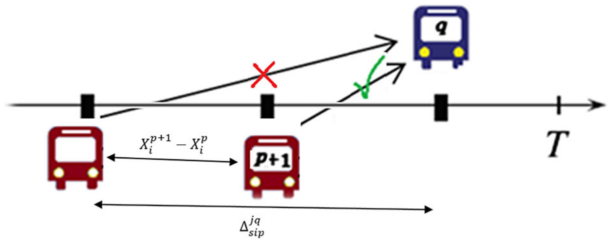



**Fig. 6** Example of a non-synchronization between the bus $p$ and $q$

In practice, the solution of the problem addressed must be proposed in a reasonable time, and part of the quality of the solution is sacrificed in the mathematical sense to speed up the resolution process. In this work, the problem is NP-hard [9]. Therefore, we used a metaheuristics approach named the multi-agent approach which offers good quality solutions in a reasonable time. The multi-agent approach exploits several aspects of multiagent systems which encourages modularity and reusability. Then, the coalition structure is intended to support robustness and facilitate the distribution, since the control is decentralized and the agents' interactions are asynchronous. Finally, cooperation and learning mechanisms contribute to the effectiveness of the optimization. The effectiveness of this kind of method is based on two mechanisms, namely, intensification and diversification. Intensification allows for good exploitation of a specific area in the search space, while diversification explores other search areas not yet visited. To solve our model we will rely on distributed artificial intelligence, and more particularly on multi-agent systems (MAS) [17,18], which allow to alternate between two mechanisms, intensification and diversification in an intelligent way (the effectiveness of a metaheuristic depends on the alternation between intensification and diversification operators). The multi-agent approach is based on a set of agents that are a set of autonomous and independent entities [19,20], and that solves the problem collectively according to a strategy. Figure 7 shows an example of a multi-agent system.

### 5.1 Approach Description

The resolution approach consists of a set of agents grouped in a coalition. Each agent is able to solve the problem using a strategy and a set of intensification and diversification operators. Agent cooperation consists of exchanging the information on the successful search strategies. The objective of this approach used is to exploit the different aspects of multi-agent systems. These aspects appear, in the distribution of the resolution and in the use of mechanisms of cooperation between agents, and then in the use of techniques of artificial learning. By the multi-agent approach it becomes easier to use the parallel programming in the resolution.

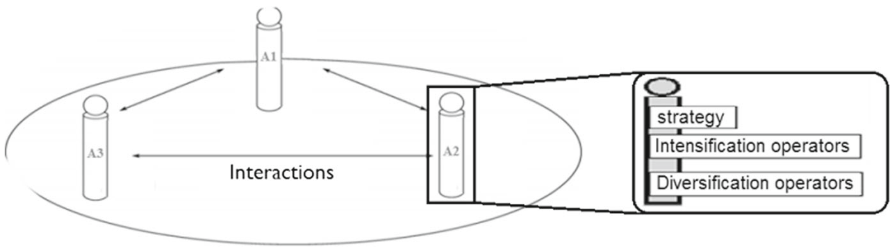



**Fig. 7** Example of a coalition in multi-agent approach

In each agent, several steps are executed taking into consideration the cooperation with other agents. We summarize these steps in Fig. 8 [21]. Then, we explain each step. The behavior of an agent is based on the operators, the decision process, and the learning mechanisms [22]. At the beginning, each agent must be initialized with a solution of the problem. A solution of the problem is coded as follows:

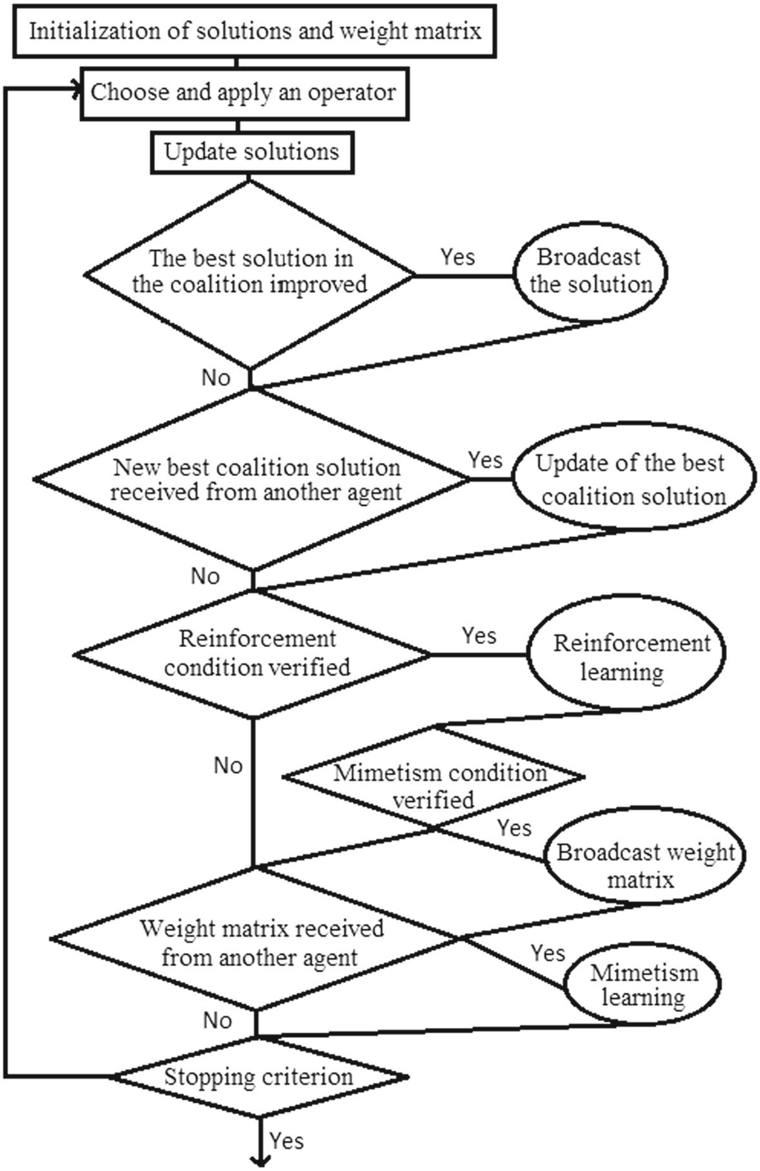
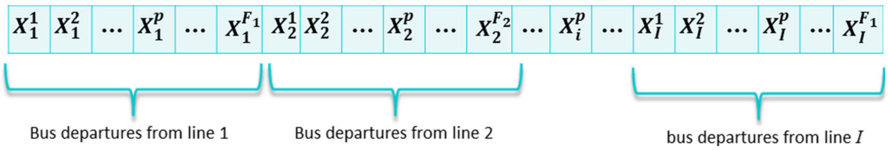



**Fig. 8** The behavior of an agent




### 5.1.1 Operators

The choice of operators is based on the strategy of each agent, and each agent has a set of operators, which allows, to diversify the search to explore all the search space and, to exploit each area of the search space. In the following section, we present two types of operators, diversification operators, and intensification operators.

**Intensification operators**

The intensification operators are mainly based on successive movements that improve the fitness in the neighborhood of an initial solution, it is an improvement based on the local search (local descent) using four neighborhood structures:

- The first neighborhood modifies a single variable (a single start) of the solution.
- The second neighborhood is defined by the modification of two consecutive starts.
- The third neighborhood is defined by the variation of all the departures of a single line in the solution.
- The last type of neighborhood is obtained by shifting a set of corresponding departures.

The local descent procedure is illustrated in the following algorithm using the neighborhoods mentioned [19].

**Algorithm 1** Local descent procedure

Step 1 **Initialization:** *find an initial solution x*
Step 2 **Repeat**
Step 3 $disp \leftarrow false$
Step 4 **Search in the neighborhood:** *find a solution* $x' \in V(x)$
Step 5 **if** $(f(x') < f(x))$ **then**
Step 6 $\quad x \leftarrow x'$
Step 7 $\quad disp \leftarrow true$
Step 8 **end if**
Step 9 **Until**$disp = False$

**Diversification operators**

The operators used in our approach are drawn from evolutionary algorithms, corresponding to the generation, mutation and crossover procedures, and performing the diversification task. This means that it allows to change the search area in the search space to avoid a local solution.

- Greedy generation
  According to the constraints (4.2), (4.3), and (4.4), the greedy generation builds the schedules by randomly selecting departures in the time window expressed by

the following expression, which represents the admissible domain of $X_i^p$.

$$[(p-1)h_i^{\min}, T-(F_i-p)h_i^{\min}] \cap [X_i^{p-1}+h_i^{\min}, X_i^{p-1}+h_i^{\max}].$$

To explain this operator, we generate an initial feasible solution of the multi-agent approach in a planning horizon $T = 30$. For example, we consider a bus network with three lines. Minimum headway, maximum headway of each line and the number of buses in each line are illustrated in Table 1.
The solution is coded as follows:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $X_1^1$ | $X_1^2$ | $X_1^3$ | $X_1^4$ | $X_2^1$ | $X_2^2$ | $X_2^3$ | $X_2^4$ | $X_3^1$ | $X_3^2$ | $X_3^3$ |

We generate the departures of the first line, and in the same way, we can generate the departures of the other lines, based on the following formula:

$$[(p-1)h_i^{\min}, T-(F_i-p)h_i^{\min}] \cap [X_i^{p-1}+h_i^{\min}, X_i^{p-1}+h_i^{\max}].$$

For $p = 1$, that is to say the first bus of the line, there is no constraint with the previous bus $(p-1)$, so we randomly take $X_1^1$ belonging to $[0, h_1^{\max}]$, for example, $X_1^1 = 2 \in [0, 13]$, and the solution is coded as follows:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | $X_1^2$ | $X_1^3$ | $X_1^4$ | $X_2^1$ | $X_2^2$ | $X_2^3$ | $X_2^4$ | $X_3^1$ | $X_3^2$ | $X_3^3$ |

For $p = 2$, $X_1^2 \in [(2-1)8, 30-(4-2)8] \cap [2+8, 2+13]$. Involves $X_1^2 \in [8, 14] \cap [10, 15]$. Therefore, $X_1^2 \in [10, 14]$, for example $X_1^2 = 13$, and the solution is coded as follows:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 13 | $X_1^3$ | $X_1^4$ | $X_2^1$ | $X_2^2$ | $X_2^3$ | $X_2^4$ | $X_3^1$ | $X_3^2$ | $X_3^3$ |

Similarly, we find $X_1^3 \in [16, 22] \cap [21, 26]$. Therefore, $X_1^3 \in [21, 22]$, for example $X_1^3 = 21$, and the solution becomes

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 13 | 21 | $X_1^4$ | $X_2^1$ | $X_2^2$ | $X_2^3$ | $X_2^4$ | $X_3^1$ | $X_3^2$ | $X_3^3$ |

For the last departure of the line, we find $X_1^4 \in [29, 30]$, for example $X_1^4 = 30$, and the solution becomes

**Table 1** Example characteristics

| | Line 1 | Line 2 | Line 3 |
|---|---|---|---|
| $h_{\min}$ | 8 | 8 | 5 |
| $h_{\max}$ | 13 | 10 | 8 |
| $F_i$ | 4 | 4 | 3 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 13 | 21 | 30 | $X_2^1$ | $X_2^2$ | $X_2^3$ | $X_2^4$ | $X_3^1$ | $X_3^2$ | $X_3^3$ |

The same procedure is repeated in the other lines, and we find the following feasible solution:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 13 | 21 | 30 | 0 | 8 | 16 | 26 | 5 | 10 | 15 |

- Crossover procedure

  The crossing procedure used is illustrated in the following steps:
  Select a departure ($p$) from each line at random, and transmit for the first $p-1$ elements (genes) from the parent (P1) to the child (E1) for each line. That is, the child (E1) keeps the same departures as (P1) for the first $p-1$ elements of the individual for each line, and for the rest we use the Algorithm 2.
  To illustrate the rule used in the algorithm we consider two parent chromosomes (P1) and (P2), with the same characteristics (number of lines, number of buses in each line), and we use the notation presented in both parents in the algorithm.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_1^1$ | $X_1^2$ | ... | $X_1^p$ | ... | $X_1^{F_1}$ | ... $X_i^p$ | ... | $X_I^1$ | $X_I^2$ | ... | $X_I^p$ | ... | $X_I^{F_I}$ | | P1 |
| $X_1^{\prime 1}$ | $X_1^{\prime 2}$ | ... | $X_1^{\prime p}$ | ... | $X_1^{\prime F_1}$ | ... $X_i^{\prime p}$ | ... | $X_I^{\prime 1}$ | $X_I^{\prime 2}$ | ... | $X_I^{\prime p}$ | ... | $X_I^{\prime F_I}$ | | P2 |

**Algorithm 2** The crossing procedure

Step 1 **Select a depart $p$ in each line from the parent P1 randomly**
Step 2 **For** *each line $i$* **do**
Step 3 **For** *each depart $k = p$ to $F_i$* **do**
Step 4 **If** *the depart selected $X_i^k$ is the first in the line* **do**
Step 5 $(X_i^k)E1 \leftarrow X_i^{\prime k}$
Step 6 **Else**
Step 7 **if** $(X_i^{\prime p} \in [(X_i^{k-1})E1 + h_i^{\min}, (X_i^{k-1})E1 + h_i^{\max}])$
Step 8 $(X_i^p)E1 \leftarrow X_i^{\prime p}$
Step 9 **Else**
Step 10 **if**$X_i^{\prime p} < (X_i^{k-1})E1 + h_i^{\min}$
Step 11 $(X_i^p)E1 \leftarrow (X_i^{p-1})E1 + h_i^{\min}$
Step 12 **Else**
Step 13 $(X_i^p)E1 \leftarrow (X_i^{k-1})E1 + h_i^{\max})$
Step 14 **End if**
Step 15 **End if**
Step 16 **End if**
Step 17 **End for**
Step 18 **End for**
Step 19 **End**

The same procedure is repeated to generate the second child, by guarding at the beginning the first $p$ departures of the second parent in the second child. For example, if we consider a bus network with three lines, two solutions have been constructed from Table 1:

– The first solution (Parent 1)

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | ↓ | | | ↓ | ↓ | | | |
| 0 | 8 | 17 | 26 | 2 | 10 | 20 | 28 | 2 | 7 | 12 |

– The second solution (Parent 2)

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | ↓ | | | ↓ | | ↓ | | |
| 2 | 13 | 21 | 30 | 0 | 8 | 16 | 26 | 5 | 10 | 15 |

For example, consider the crossover point $p = 3$ in the second line. The child (E1) keeps the same departures as (P1) for the first $p - 1$ elements of the line (parent 1), and we have

$$X_2^{\prime 3} = 16 \notin [10 + 8, 10 + 10], \text{ and } \quad X_2^{\prime 3} = 16 < 18.$$

So, $(X_2^3)$E1=18.

For the other departures, we use the procedure explained above, and we get the following first child:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 8 | 17 | 30 | 2 | 10 | 18 | 26 | 5 | 10 | 15 |

With the same principle we find the second child:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 13 | 21 | 26 | 0 | 8 | 18 | 28 | 2 | 7 | 12 |

– Mutation procedure:
We randomly choose a departure index $p$ in each line in the solution. The modification is chosen randomly in the following definition interval:

$$[X_i^{p-1} + h_i^{\min}, X_i^{p-1} + h_i^{\max}] \cap [X_i^{p+1} - h_i^{\min}, X_i^{p+1} - h_i^{\max}].$$

### 5.1.2 Decision Process

The decision process differentiates efficient operators. It allows to select a sequence of operators to apply as a diversification-intensification cycle defined by the choice of a diversification operator followed by an intensification phase ending when the current solution is a local optimum on all the neighborhood structures considered. The selection mechanism implemented is based on the condition/action weight matrix initialized by a parameter $\alpha$. The effective choice of an operator is performed by a roulette wheel selection principle.

The matrix of weights is presented in Fig. 9, where

- $(i_1, i_2, \cdots, i_p)$, intensification operators;
- $(d_1, d_2, \cdots, d_p)$, diversification operators;
- $c_0$: condition activated after the application of a diversification operator;
- $c_{n-1}$: activated when the current solution is a local optimum in all neighborhoods used.

Each condition $c_i$ is activated after the application of an operator in $c_{i-1}$.

The selection of an operator $o_j$ in the condition $c_j$ is computed using the following formula:

$$P(o_j/c_i) = \frac{\omega_{ij}}{\sum_{k=1}^{k=m} \omega_{ik}},$$

where

- $C : (c_i)_{i=1,\cdots,n}$: set of conditions;
- $O : (o_j)_{j=1,\cdots,m}$: set of operators;
- $W : (\omega_{ij})_{i=1,\cdots,n,j=1,\cdots,m}$: weight matrix.

### 5.1.3 Learning Mechanisms

After the application of a cycle of operators, we used two learning mechanisms to improve the choice of operators by adjusting the weight matrix of the decision process. Two learning mechanisms are used by reinforcement and by mimetism.

**Reinforcement Learning Procedure**

The principle of reinforcement learning is to increase the weights associated with beneficial operators after the application of an intensification/diversification cycle. Two types of modification are defined when only the best solution of the agent is improved, the value $\sigma_1$is applied, and if the best solution in the coalition is improved, the value $\sigma_2$ is applied [24,25].

The reinforcement is performed using the formula:

$$\omega_{ij} = \omega_{ij} + \sigma,$$

Fig. 9 Decision matrix [23]

| | $i_1$ | $i_2$ | $\cdots$ | $i_p$ | $d_1$ | $d_2$ | $\cdots$ | $d_a$ |
|---|---|---|---|---|---|---|---|---|
| $c_0$ | $\alpha$ | $\alpha$ | $\cdots$ | $\alpha$ | 0 | 0 | $\cdots$ | 0 |
| $c_1$ | $\alpha$ | $\alpha$ | $\cdots$ | $\alpha$ | 0 | | $\cdots$ | 0 |
| | $\alpha$ | $\alpha$ | | | | | | |
| $\vdots$ | $\vdots$ | | $\ddots$ | $\vdots$ | $\vdots$ | | $\ddots$ | $\vdots$ |
| | | | $\alpha$ | $\alpha$ | | | | |
| $c_p$ | $\alpha$ | | $\cdots$ $\alpha$ | $\alpha$ | 0 | | $\cdots$ | 0 |
| $c_{n-1}$ | 0 | | 0 | 0 | $\alpha$ | $\alpha$ | $\cdots$ | $\alpha$ |

where

- $(c_i, o_j)$: experience to reinforce;
- $\omega_{ij}$: weight related to the experience;
- $\sigma : \{\sigma_1, \sigma_2\}$: learning factors.

In the following example, we explain the procedure of reinforcement learning.

***Example* 5.1** After the application of an intensification/diversification cycle, the diversification operator (o5), intensification operators (o2, o1, o3, o4), the best solution in the agent improved are as shown in the right side of Fig. 10. So, the weights of these operators are modified to favor the selection of the operators in the next selections in the same conditions. The left side of Fig. 10 illustrates the reinforcement with $\sigma = 2$ of the weight matrix initialized with $\alpha = 1$. For instance, if condition c1 is activated after the reinforcement, then the operator o1 has a probability of 50% to be selected, compared to 25% before the reinforcement.

**Mimetism Learning Procedure**

If the best coalition solution is improved, the agent broadcasts its weight matrix to the other agents of the coalition. Then, agents that receive the weight matrix apply a procedure of mimetism learning using the following formula:

$$W_B = (1 - \rho) W_B + \rho W_A,$$

where

- $W_A$: weight matrix of the imitator agent;
- $W_B$: weight matrix of the imitated agent';
- $\rho$: mimetism rate.

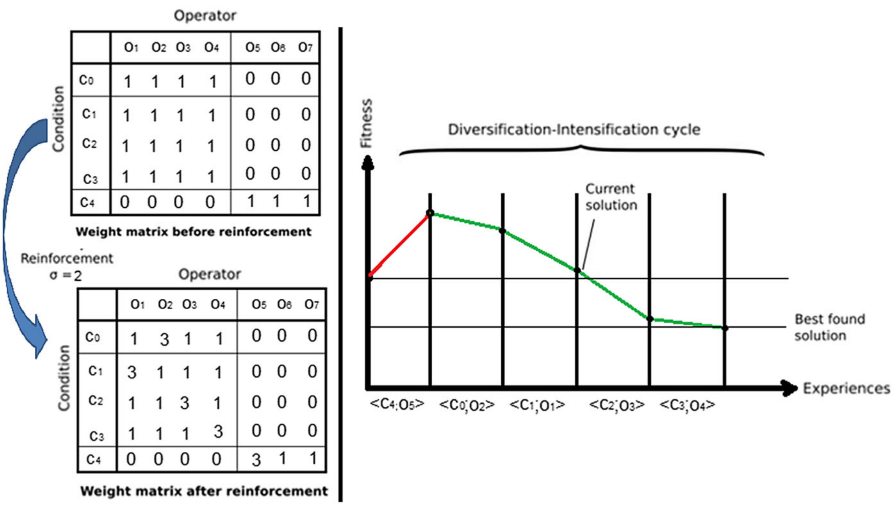



**Fig. 10** Example of reinforcement learning

The behavior of an agent is summarized in the following algorithm:

**Algorithm 3** The behavior of an agent

Step 1 **Initialization of solutions**
Step 2 **Initialization of the weight matrix**
Step 3 **While** *stopping criterion is not reached* **do**
Step 4 *Choose an operator*
Step 5 *Apply an operator*
Step 6 *Update solutions*
Step 7 **If** *the best solution in the coalition improved* **then**
Step 8 *Broadcast the solution*
Step 9 **End if**
Step 10 **If** *new best coalition solution received from another agent* **then**
Step 11 *Update of the best coalition solution*
Step 12 **End if**
Step 13 **If** *reinforcement condition* **then**
Step 14 *Reinforcement learning*
Step 15 **If** *mimetism condition* **then**
Step 16 *Broadcast weight matrix*
Step 17 **End if**
Step 18 **End if**
Step 19 **If** *weight matrix received from another agent* **then**
Step 20 *Mimetism learning*
Step 21 **End if**
Step 22 **End**

## 6 Experimental Study

The multi-agent approach applied to our model was implemented using C++ language on a Windows 8 Intel (R) Core i3 (4 CPUs) at 3.3 GHz, with 4.00 GB of RAM. In order to compare the models performance with an existing model reported in the literature, we base our experimental study on two cases: Example 2 explains how the model works, and Example 3 compares its performance with another model to solve the same problem. We use the instance generation scheme used in Fouilhoux et al. [10]. We fix the deviations of scheduled times since the instances are not real data.

***Example* 6.1** In this example, the proposed model is applied to a real-life problem which was introduced in Eranki [5]. The example is a network with three nodes and six routes, and the data are presented in Table 2.

**Table 2** Example characteristics

| $H_{min}$ | $H_{max}$ | $F_i$ | $W_i$ |
|---|---|---|---|
| 14 | 20 | 12 | 5 |

The travel times on each link, the common stations between the lines of the network, and the starting stations of each line are shown in Fig. 11 for example, the bus route 4 is represented by a continuous line passing through its departure and stations 2 and 3, and the time needed to move from station 2 at station 3 is 7 min.

The results obtained by our method are the departure times of each bus in the planning horizon as shown, in Table 3.

By solving a real-life problem with small size, we were able to find better results than the classical mining of synchronization, which shows that with our model, we can find other better solutions, as we explain in Example 3.

***Example* 6.2** We used the instance generation scheme used in Ibarra-Rojas and Rios-Solis (2016). We generated 12 instance types to clarify the performance of our model, illustrated in Table 4, where $\mid I \mid$ is the number of lines and $\mid B \mid$ is the number of synchronization nodes where a different pair of lines should be synchronized. The planning duration of $T = 120$ min.

All the instances have the following common characteristics:

- The frequency $F_i$ is randomly generated in [4, 10] for each line $i$.
- The duration $t_s^i$ to move from the start of line $i$ to the synchronization zones $s$ is randomly generated in [20, 60] for each line $i$ in the network.
- The walking time for each correspondence $s$ is randomly generated in [3, 5].

**Fig. 11** Real life problem

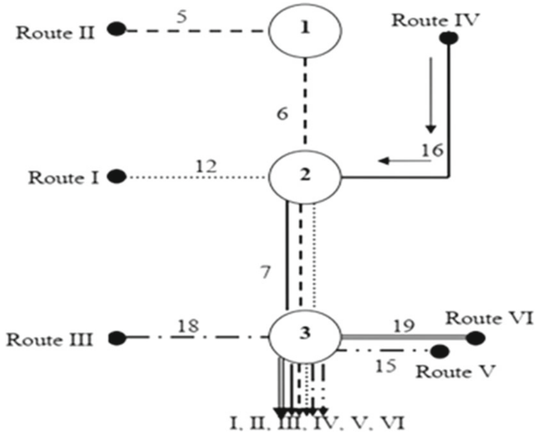



**Table 3** Computational results for real life problem

| Criterion | No planning | Classical mining of synchronization | Proposed model |
|---|---|---|---|
| Total transfer waiting time (min) | 2951 | 2430 | 2053 |
| Average transfer waiting time (min) | 8.107 | 7.617 | 5.211 |
| Number of synchronization | 364 | 319 | 394 |

**Table 4** Instance characteristics

| instance | $\mid I \mid$ | $\mid B \mid$ | $\lambda_i$ |
|---|---|---|---|
| A1 | 05 | 3 | 06 |
| A2 | 05 | 3 | 10 |
| A3 | 15 | 3 | 06 |
| A4 | 15 | 3 | 10 |
| A5 | 20 | 5 | 06 |
| A6 | 20 | 5 | 10 |
| A7 | 25 | 5 | 06 |
| A8 | 25 | 5 | 10 |
| A9 | 35 | 8 | 06 |
| A10 | 35 | 8 | 10 |
| A11 | 40 | 8 | 06 |
| A12 | 40 | 8 | 10 |

– The maximum (resp. minimum) headway $h_{\max} = \frac{T}{F_i} + \lambda_i$ (resp. $h_{\min} = \frac{T}{F_i} - \lambda_i$), with $\lambda_i$ is the flexibility parameter.

After the generation of the instances according to the parameters above, we randomly generated solutions for each instance, and for each solution we calculated the average waiting time in each correspondence station. Then, we solved the same instance using the classical mining of synchronization that we explained in our article. The results that we obtained are shown in Table 5. To clarify our contribution, we solved our mathematical model with the multi-agent approach and the results that we obtained are shown in the following table.

**Table 5** Results example 2

| Instance | No planning | Classical mining of synchronization | New mining of synchronization |
|---|---|---|---|
| A1 | 04 min 18 s | 04 min 48 s | 03 min 20 s |
| A2 | 03 min 27 s | 03 min 48 s | 03 min 04 s |
| A3 | 05 min 06 s | 04 min 35 s | 04 min 05 s |
| A4 | 05 min 12 s | 04 min 27 s | 03 min 36 s |
| A5 | 09 min 27 s | 07 min 54 s | 06 min 52 s |
| A6 | 09 min 32 s | 07 min 54 s | 07 min 07 s |
| A7 | 08 min 49 s | 07 min 26 s | 07 min 21 s |
| A8 | 09 min 00 s | 07 min 23 s | 06 min 43 s |
| A9 | 14 min 33 s | 12 min 09 s | 11 min 29 s |
| A10 | 14 min 45 s | 11 min 32 s | 11 min 20 s |
| A11 | 14 min 57 s | 13 min 29 s | 13 min 05 s |
| A12 | 15 min 08 s | 13 min 11 s | 12 min 39 s |

We compared the average waiting time in each synchronization node obtained by solving the 12 generated instances. The results are shown in Fig. 12 where the horizontal axis represents the different instances with $\lambda_i = 6$, and the vertical axis represents the waiting times in minutes. The green color in each instance represents the waiting time in the case of non-planning, the blue indicates the waiting time in the case of classical synchronization, and our results, obtained by our new model are presented with the red color in the graph.

From this figure, the following results can be observed.

1. The waiting time in each correspondence station decreased for each instance in our model compared to other cases, which shows that the bus timetabling problem with the new definition of synchronization gives a better planning of bus departure times. For example, the results found with our model in the instance A5 decrease the waiting time from 7 min 54 s to 6 min 52 s.
2. A particular case in the instance A1, which is the solution in case "No planning", the solution found is better than the solution in the case "Classical mining of synchronization". This can be explained by the random generation of the solution. More than that, there is a probability of finding the optimal solution in the small istances, but it decreases depending on the size of the network.
3. The waiting time obtained depends on the instance types. That is, when the number of lines increased, the average waiting time increased. For the following reason, finding or approaching the optimal solution in small instances is easy compared to larger instances, the method used in the resolution allows to approach the optimal solution in a resonable time. So if the instance is bigger, the search space is wide enough and there is a probability to find a bad solution in a reasonable time.

We can see in Fig. 13 that the result obtained in the case of $\lambda_i = 10$ minimizes the average waiting time in the correspondence station better than the case of $\lambda_i = 6$ in each instance, which shows that if the headway variation window is greater, then the

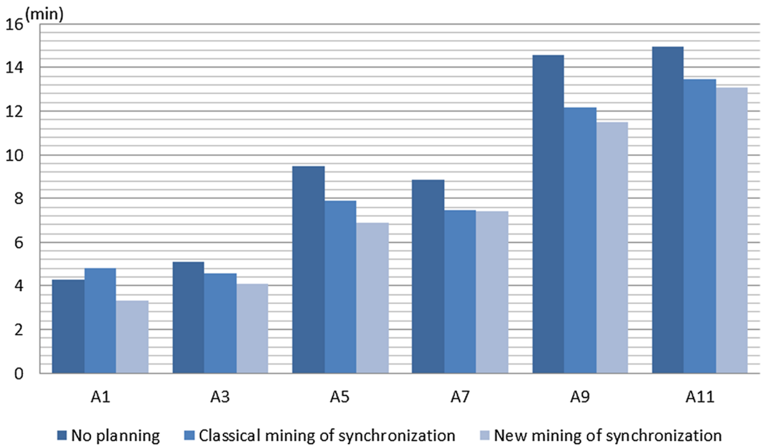



**Fig. 12** Graphical representation of the waiting time according to the instances

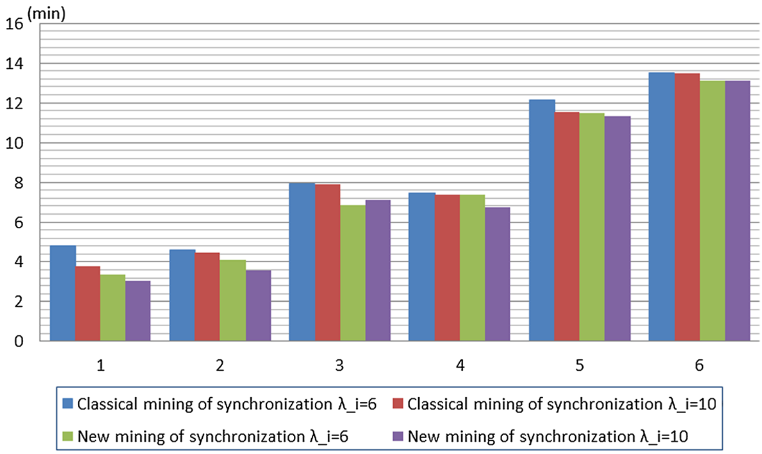



**Fig. 13** Different headway results

waiting time decreases. The reason for this is that the admissible domain of $X_i^p$ (the departure time) becomes wide.

## 7 Conclusions

In this paper, we studied one of the major problems in urban transport, which is the waiting time in bus networks. Indeed, the minimization of waiting time in the network is among the great factors that influence on the quality of service. For that we presented a new model for the scheduling problem in transit networks in order to decrease the passengers transfer waiting time in correspondence zones. In the literature, the works that deal with this problem are based on synchronization between buses, and for that we have presented a mathematical model based on the synchronization between buses. Our model gives a new definition to the synchronization between buses in order to better minimize the waiting time in the correspondence zones. We tested our model with a real-life problem (small size), and to show the efficiency of our model we generated different instances and compared the results obtained with the results using the classical definition of synchronization.

The complexity of the problem is mainly due to the use of a large search space. Thus, to solve our model, we rely on distributed artificial intelligence. In this way, we adapted an approach based on multi-agent systems.

The follows are some extension suggestions for future research

- In our model, we assumed that bus capacity is unlimited, but in real life, each bus has a maximum capacity. Hence the bus capacity can be considered.
- Another study that can be done is to analyze the number of buses needed to make the waiting time null.

## References


[1] Jansen, L.N., Pedersen, M.B., Nielsen, O.A.: Minimizing passenger transfer times in public transport timetables. In: 7th Conference of the Hong Kong Society for Transportation Studies, pp. 229–239. Transportation in the information age, Hong Kong (2002)
[2] Guihaire, V., Hao, J.K.: Transit network design and scheduling: a global review. Transp. Res. Part A Policy Pract. **42**(10), 1251–1273 (2008)
[3] Daduna, J.R., Voß, S.: Practical experiences in schedule synchronization. In: Daduna, J.R., Branco, I., Paixao, J.M.P. (eds.) Computer-Aided Transit Scheduling, pp. 39–55. Springer, Berlin (1995)
[4] Ceder, A., Golany, B., Tal, O.: Creating bus timetables with maximal synchronization. Transp. Res. Part A Policy Pract. **35**(10), 913–928 (2001)
[5] Eranki, A.: A model to create bus timetables to attain maximum synchronization considering waiting times at transfer stops (2004)
[6] Cevallos, F., Zhao, F.: Minimizing transfer times in public transit network with genetic algorithm. Transp. Res. Rec. J. Transp. Res. Board **1971**, 74–79 (2006)
[7] Shrivastava, P., Dhingra, S.: Development of coordinated schedules using genetic algorithms. J. Transp. Eng. **128**(1), 89–96 (2002)
[8] Zhigang, L., Jinsheng, S., Haixing, W., Wei, Y.: Regional bus timetabling model with synchronization. J. Transp. Syst. Eng. Inf. Technol. **7**(2), 109–112 (2007)
[9] Ibarra-Rojas, O.J., Rios-Solis, Y.A.: Synchronization of bus timetabling. Transp. Res. Part B Methodol. **46**(5), 599–614 (2012)
[10] Fouilhoux, P., Ibarra-Rojas, O.J., Kedad-Sidhoum, S., Rios-Solis, Y.A.: Valid inequalities for the synchronization bus timetabling problem. Eur. J. Oper. Res. **251**(2), 442–450 (2016)
[11] Feng, X., Zhu, X., Qian, X., Jie, Y., Ma, F., Niu, X.: A new transit network design study in consideration of transfer time composition. Transp. Res. Part D Transp. Environ. **66**, 85–94 (2019)
[12] Ibarra-Rojas, O.J., Muñoz, J.C., Giesen, R., Knapp, P.: (2019) Integrating frequency setting, timetabling, and route assignment to synchronize transit lines. J. Adv. Transp. (2019). https://doi.org/10.1155/2019/9408595
[13] Guihaire, V.: Modélisation et optimisation pour le graphicage des lignes de bus. Ph.D. Thesis, Université d'Angers (2009)
[14] TRB: Transit Capacity and Quality of Service Manual (TCRP 100), 2nd edn. Washington (2003)
[15] TRB: Traveller Response to Transportation System Changes: Transit Scheduling and Frequency (TCRP 95). Washington (2004)
[16] Pratt, R.H., Coople, J.: Traveler response to transportation system changes. Media Info, p. 408 (1981)
[17] Jacques, F.: Les systèmes multi-agents, vers une intelligence collective, p. 322. InterEditions, Paris (1995)
[18] Laichour, H.: Modélisation multi-agent et aide à la décision: application à la régulation des correspondances dans les réseaux de transport urbain. Ph.D. Thesis, Lille 1 (2002)
[19] Meignan, D.: Une approche organisationnelle et multi-agent pour la modélisation et l'implantation de métaheuristiques, application aux problèmes d'optimisation de réseaux de transports. Ph.D. Thesis, Université de Technologie de Belfort-Montbeliard (2008)
[20] Zgaya, H.: Conception et optimisation distribuée d'un système d'information d'aide à la mobilité urbaine: Une approche multi-agent pour la recherche et la composition des services liés au transport. Ph.D. Thesis, Ecole Centrale de Lille (2007)
[21] Özcan, E., Bilgin, B., Korkmaz, E.E.: A comprehensive analysis of hyper-heuristics. Intell. Data Anal. **12**(1), 3–23 (2008)
[22] Meignan, D., Créput, J.C., Koukam, A.: An organizational view of metaheuristics. In: First International Workshop on Optimisation in Multi-Agent Systems, AAMAS, vol. 8, pp. 77–85 (2008)
[23] Meignan, D., Créput, J., Koukam, A.: Un framework organisationnel pour la conception et l'implantation multi-agent de métaheuristiques (2008)
[24] Meignan, D., Créput, J.C., Koukam, A.: A coalition-based metaheuristic for the vehicle routing problem. In: Evolutionary Computation, 2008. CEC: IEEE World Congress on Computational Intelligence, IEEE Congress on, IEEE pp. 1176–1182 (2008)
[25] Meignan, D., Koukam, A., Créput, J.C.: Coalition-based metaheuristic: a self-adaptive metaheuristic using reinforcement learning and mimetism. J. Heurist. **16**(6), 859–879 (2010)