European Journal of Operational Research 246 (2015) 772–786

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

# High-speed railway scheduling based on user preferences

José Luis Espinosa-Aranda[*], Ricardo García-Ródenas, María del Carmen Ramírez-Flores, María Luz López-García, Eusebio Angulo

*Universidad de Castilla La Mancha, Escuela Superior de Informática, Paseo de la Universidad, 4, Ciudad Real 13071, Spain*

ARTICLE INFO

*Article history:*
Received 24 June 2014
Accepted 19 May 2015
Available online 29 May 2015

*Keywords:*
Timetabling
High-speed railway
Constrained nested logit model
Utility theory
Metaheuristics

ABSTRACT

This paper proposes an optimization model for high-speed railway scheduling. The model is composed of two sub-models. The first is a discrete event simulation model which represents the supply of the railway services whereas the second is a constrained logit-type choice model which takes into account the behaviour of users. This discrete choice model evaluates the attributes of railway services such as the timetable, fare, travel time and seat availability (capacity constraints) and computes the High-Speed railway demand for each planned train service.

A hybridisation of a Standard Particle Swarm Optimisation and Nealder–Mead methods has been applied for solving the proposed model and a real case study of the High-Speed corridor Madrid–Seville in Spain has been analysed. Furthermore, parallel computation strategies are used to speed up the proposed approach.

## 1. Introduction

European Union competition policy is promoting an ongoing liberalisation process in the rail market and the setting up of an EU air market. In this single transport market, the low-cost air carriers and the high-speed railway (HSR) operators are competitors on routes that are interchangeable within a certain distance range, especially in point-to-point routes. Currently HSR systems are expanding their rail networks in the EU and incrementing their demand share. However, in order to meet the new scenario of competition between transport operators, the rail industry should focus the railway service on addressing passenger needs.

*The Train Timetabling Problem* (TTP) consists of the establishment of a set of conflict-free train routes with respect to given train requirements. TTP has received a great deal of attention in the scientific literature in recent years. The literature focuses on minimizing parameters related to train company operations (operations-centric models) but so-called passenger-centric models have also been developed but they are much sparser. These analytical models are crucial in improving the competitiveness of the rail industry.

HSR systems are an emergent technology whose singular features have not yet been widely discussed in the current literature on TTP modelling. The departure time choice, travel time, the pricing regime and the competition of these systems with the traditional or low-cost air, rail and private vehicles should be taken into account in the TTP models. Motivated by this need, this paper proposes a high-speed passenger-centric TTP model that uses a high-speed demand forecasting model to learn the behaviour of passengers with respect to the attributes (departure time, travel time, ticket cost and travel comfort) of the proposed timetable.

The remainder of this paper is structured as follows. Section 2 reviews related research papers. Section 3 describes the proposed HSR–TTP model in detail. Section 4 explains the algorithms used for solving this model, in Section 5 several computational experiments are reported, and finally Section 6 concludes with a discussion of our findings and future work.

## 2. Literature review

TTP has been addressed widely in the literature in the last thirty years. A rough taxonomy of TTP may be classified, see Cacchiani and Toth (2012), according to application: (i) freight or passenger transportation, (ii) railway network or single one-way line (corridor) linking two major stations with intermediate stations, (iii) cyclic (periodic) or non-cyclic timetables and (iv) objective function. In this section we focus exclusively on HSR–TTP approaches.

Cascetta and Coppola (2014) consider that a high-speed-generated demand obeys three main factors:

(1) The *diverted demand*, which represents the choice of a passenger between other means of transport (plane, car, other rail services, etc.) and HSR.
(2) The *induced demand*, which depends directly on the characteristics of the HSR services offered (ticket cost, travel time,

[*] Corresponding author. Tel.: +34 926 29 53 00x6222.
*E-mail address:* JoseL.Espinosa@uclm.es (J.L. Espinosa-Aranda).

http://dx.doi.org/10.1016/j.ejor.2015.05.052

**Table 1**
Review of HSR disaggregate models (source: Cascetta and Coppola (2014)).

| | | Frequency-based | Schedule-based |
|---|---|---|---|
| **Multi-modal** | Multi rail service | Ben-Akiva et al. (2010) | Cascetta and Coppola (2012), Cascetta and Coppola (2014), Espinosa-Aranda et al. (2014) |
| | Single rail service | Román et al. (2007), Yao and Morikawa (2005), Fröidh (2008) | |
| **Mono-modal** | Multi rail service | Couto and Graham (2008), Hsu and Chung (1997) | |
| | Single rail service | Urban case | Urban case |

timetable, etc.), or indirectly due to modifications of the travellers' mobility or lifestyle choices.

(3) The *economy-based demand growth*, which is linked to the trends of the economic system, considering that people travel more when they are wealthy.

Diverted demand and directly induced demand are considered as *endogenous* factors of the HSR system. Indirectly induced demand and economy-based demand growth are considered as *exogenous* factors of the HSR system. Numerous HSR demand-forecasting models have been developed to predict rail passenger flows. Some of these involve short-term forecasting (Jiang, Zhang, & Chen, 2014) and their aim is to provide daily user estimates that account for day-to-day demand variations in the near future. These models have been applied to problems of revenue management, improvement of facilities, operation mode, etc. and predictions are made considering the HSR system as a black box. A model which tries to solve the TTP for HSR systems must be capable of forecasting high-speed rail demand as a function of the endogenous factors of the system.

HSR demand forecasting models can be classified as *aggregate* or *disaggregate*. Aggregate models are based on aggregate demand elasticity values and make use of large data sets obtained from ticket sales and surveys. They are useful for rail demand growth predictions (see Wardman, 2006) or policy simulation, such as the estimation of the effects of $CO_2$ emissions taxes on the demands for airline and HSR travel (see Fu, Oum, & Yan, 2014) or to forecast the demand for a proposed new HSR service (see Börjesson, 2014). The fundamental limit on incorporating these models into TTP models is that they cannot simulate flows on individual rail segments or trains. Disaggregate models consider the individual as the basic unit of observation, dividing the passengers into different types depending on different factors. These models are consistent with travel choice theory using data at the level of individual travellers.

Cascetta and Coppola (2014) creates a taxonomy of HSR disaggregate demand forecasting models (see Table 1). The two essential classification criteria which apply are the type of transport network (*multi-modal* or *mono-modal*) and the way the range of services is represented (*frequency-based* or *schedule-based*). This classification is very useful for properly situating the model given in this study in its place in the literature. This is line of thought followed here.

*Mono-modal* models are focused on a specific transport mode and forecast the demand based on the operational improvement of this mode (for some examples see Couto and Graham (2008) and Hsu and Chung (1997)). These models are typically conceived to represent the behaviour of the passengers in an urban context. *Multi-modal* models include the competition between HSR and other modes of transports available for the same trip. Most of these models focus on the competition between HSR and air transportation (see Fröidh, 2008; Park & Ha, 2006; Román, Espino, & Martín, 2007), some include cars (see Mandel, Gaudry, & Rothengatter, 1997; Martín & Nombela, 2007; Yao & Morikawa, 2005) and very few add the competition between HSR operators (see Ben-Akiva, Cascetta, Coppola, Papola, & Velardi, 2010; Cascetta & Coppola, 2012).

The frequency-based approaches are focused on defining a frequency for each type of service over given time intervals. Greater flexibility in the modelling is given by scheduling-based approaches which allow scheduled services to be represented as individual trips following a timetable (see Wilson & Nuzzolo, 2004). The majority of the models presented in the literature to forecast HSR demand follow a frequency-based demand approach. The choice of one or other approach conditions the output of the TTP model, directing it towards establishing frequency o defining a timetable. For low-frequency transport systems the schedule-based approach reflects user choice in a specific service better.

Schedule-based studies like Espinosa-Aranda, García-Ródenas, López-García, and Angulo-Herrera (2014) where a constrained nested logit model is proposed with the objective of predicting the demand in an HSR system are scarce. This model considers the timetable, price of the ticket, travel time and capacity of the trains in the selection process. These authors use *Reproducing Kernel Hilbert spaces* to consider dynamic utilities, which is an essential issue for modelling the departure time choice.

The other main issue in TTP is the modelling of the railway services supplied. Traditionally to represent this problem several types of constraints like block capacity constraints, flow constraints, priority constraints, train service capacity constraints and rolling stock constraints have been considered (see Cordeau, Toth, & Vigo, 1998). HSR systems have received a great deal of investment which eases the operational control of these systems allowing the simplification of the traditional regional or intercity railway systems modelling. This paper proposes a discrete event simulation model for representing the supply of the network. This approach has been applied successfully in transportation (see Espinosa-Aranda & García-Ródenas, 2012; Sánchez-Rico, García-Ródenas, & Espinosa-Aranda, 2014). Furthermore, it is possible to extend the HSR model by changing the proposed supply model with previous approaches defined in the literature, such as methods based on the alternative graph model (see Corman, D'Ariano, Pacciarelli, & Pranzo, 2010; D'Ariano, 2008; D'Ariano, Pacciarelli, & Pranzo, 2008; Espinosa-Aranda & García-Ródenas, 2013), or including real-time optimisation methods for rescheduling rolling stock (see Almodóvar & García-Ródenas, 2013).

### 2.1. Summary and contributions of this paper

The proposed HSR–TTP approach can be classified as a disaggregate *multi-modal* schedule-based model and focuses on the endogenous factors (travel time, fare and departure time) associated with HSR passenger behaviour. The contributions of this paper can be summarized as follows:

- A novel HSR–TTP model is proposed. The model is able to accommodate the following two key features: (i) the interaction between demand and supplied capacity, explicitly considering the effect that a given set of services may be at capacity has on user choice and (ii) competition between means of transport (low-cost air carriers, other railway (HSR) operators or private car).
- An optimisation framework for the HSR–TTP model is analysed. The HSR–TTP model presents a bilevel structure which integrates a simulation approach (supply-side modelling) and an optimisation model (demand model) at lower level. The resulting model is implicitly defined and exact methods become inapplicable. Moreover, the evaluation of the objective function is expensive. This model poses a special challenge, since existing heuristic methods

**Table 2**
Complete notation used.

| | |
|---|---|
| Sets and indices | |
| $s$: | Railway service. It is the route of a train between two end stations |
| $\mathcal{S}$: | Set of services of the railway network |
| $i, j$: | Stations |
| $\mathcal{J}$: | Set of stations of the railway network |
| $k, k'$: | Trains |
| $\mathcal{K}$: | Set of trains on the railway network |
| $w$: | Origin-destination demand pair |
| $\mathcal{W}$: | Set of origin-destination demand pairs |
| $\mathcal{S}_W$: | Set of services which could cover demand $w$ |
| $\mathcal{L}$: | Set of types of user |
| $\ell$: | A type of user |
| $\mathcal{E}$: | Set of events of the railway network |
| Variables | |
| $e$: | Event in the railway network |
| $T_s$: | Departure time of service $s$ at the initial station |
| $\mathbf{T}$: | Vector of departure times $(\cdots, T_s, \cdots)$ |
| $T^a_{sj}$: | Instant in which a service $s$ arrives at a station $j$ |
| $T^d_{sj}$: | Instant in which a service $s$ departs from station $j$ |
| $\mathbf{T}^d$: | Vector $(\cdots, T^d_{s,j}, \cdots)$, i.e. the full free-conflict timetable |
| $\delta_e$: | Instant associated with event $e$ |
| $\tau^a_j$: | Instant when the last train, with respect to the current simulation clock, arrives at station $j$ |
| $\tau^d_j$: | Last programmed departure instant from station $j$ before the arrival of the next service at station $j$ with respect to the current simulation clock |
| $d_{ij}$: | Dwell time of railway track segment $i \rightarrow j$ |
| $c^k_{ij}$: | Travel time of the train $k$ in the track segment $i \rightarrow j$ |
| $p_{sj}$: | Stop time of service $s$ in station $j$ |
| $\widetilde{T}_{s'}$: | Real instant when service $s'$ starts |
| $T_{s'}$: | Planned initial time of service $s'$ |
| $\delta_s$: | End time of service $s$ |
| $r_{ss'}$: | Rolling stock time for getting the train ready for service $s$ until it can start service $s'$ |
| $\widehat{g}^\ell_W$: | Number of potential passengers type $\ell$ for demand $w$ |
| $g^{a\ell}_{\omega,s}$: | Number of passengers for demand $\omega$ who use the service $s$ |
| $\mathbf{g}^a$ | Vector of disaggregated HRS demand$(\cdots, g^{a\ell}_{\omega,s}, \cdots)$ |
| $V^{a\ell}_{\omega,s}$: | Utillity of the service $s$ to make the trip $\omega$ for user type $\ell$ |
| $\mathbf{V}$: | Vector of utilities, i.e. $(\cdots, V^{a\ell}_{\omega,s}, \cdots)$ |
| $b_{\omega,s}$: | Price of the ticket for demand $\omega$ using service $s$ |
| $K_s$: | Capacity of service $s$ |
| $\widehat{K}_s$: | Effective capacity of service $s$ |
| $\widehat{\mathbf{K}}$ | Vector of effective capacities, i.e. $(\cdots, \widehat{K}_s, \cdots)$ |
| $\pi_s$: | Operating costs of service $s$ |
| $\widehat{\pi}_s$: | Effective operating costs of service $s$ |

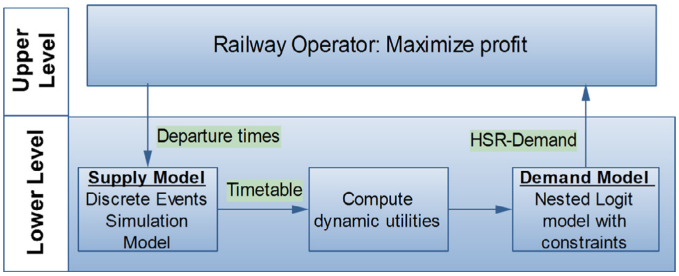
Upper Level
Railway Operator: Maximize profit
Lower Level
Departure times
HSR-Demand
Supply Model
Discrete Events Simulation Model
Timetable
Compute dynamic utilities
Demand Model
Nested Logit model with constraints

**Fig. 1.** HSR–TTP timetable model based on user behaviour.

often require more function evaluations than can be comfortably afforded. In this paper, the way to address this challenge is to use parallelisation strategies in an accelerated Particle Swarm Optimisation algorithm. We have shown in a real case that this strategy is suitable from the point of view of computational cost and of global optimisation search.

## 3. A model for HSR scheduling

This section sets out a TTP model focused on the HSR operator's profits. This model, whose structure is depicted in Fig. 1, optimizes the timetable, taking into account the behaviour of the passengers.

This HSR–TTP model comprises three modules

(1) *Supply model*. This model represents the movements of trains on the HS railway network.
(2) A model for computing the *utilities* of the scheduled services as a function of their attributes.
(3) *HSR demand model*. This model defines the choices of the users on the HSR network.

For convenience, the list of variables, sets and indices used is shown in Table 2.

### 3.1. Supply model

The set of constraints defined in this section models railway operations and overtaking between trains. A discrete event model is formulated to obtain the complete timetable assuming that initial times, speeds and stop times at stations are known. Some assumptions are made about the generic HSR system

**Assumption 1.** Double-tracked line. Each section is separated by nodes (stations) and all the trains in the same section must travel in the same direction.

**Assumption 2.** The route of each service is fixed and defined by a sequence of stations. It is not necessary to stop at all stations. A function $j = J(s, m)$ which indicates the $m$th station visited by service $s$ is assumed to be known. Also a dummy station $j_F$, which represents the end of the service, is defined.

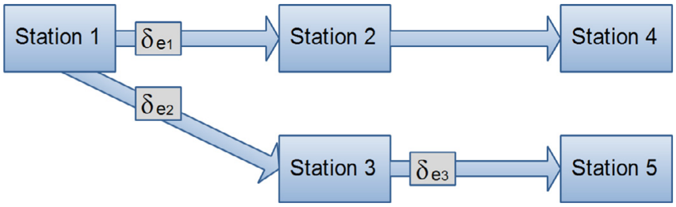

**Fig. 2.** Queue system represented as events.

**Assumption 3.** The speed of each train is fixed if there is no type of conflict with another train. In case of conflict, the following train must reduce its speed maintaining the minimum dwell time defined by controllers in each track segment.

**Assumption 4.** A train which does not stop in a station could overtake a train stopped in this station. This constraint allows the system to operate express services which do not stop in each station they go through, contributing to the improvement in the performance of the system.

Two of the most important factors in the competitiveness of HSR systems are reduced travel times and punctuality. Because of this the only time considered to be added to the travel time of each train is the necessary stop time in stations, so the following assumptions are made:

**Assumption 5.** A train can stop only the time necessary for passenger alighting/boarding, without considering the possibility of connections between trains.

**Assumption 6.** There is no priority between trains reaching a station at practically the same time, following a *First Come First Served* (FCFS) policy.

Thus the essential variables of the problem are the departure times of each train from their initial station. The supply model aims to determine the complete timetable of the HSR network.

The supply model receives the train departure times and simulates their behaviour taking into account the operational constraints. Finally the complete conflict-free timetable is calculated.

### 3.1.1. *Dwell time and overtaking constraints*

We denote by $T_s$ the start time of service $s \in \mathcal{S}$ and we call

$$T = (T_1, \cdots, T_n)$$

the vector which contains the initial departure times of planned services, where $n$ is the cardinal of the set $\mathcal{S}$.

Each train $k \in \mathcal{K}$ is constrained to a feasible number of services in a given period. To describe the model, denote as $s = S(k, m)$ a function capable of obtaining the $m$th service, that is $s$, of train $k$.

**Definition 1.** An **event** $e$ represents the instant in which a train $k$ performing service $s$ leaves a station $j$ and is defined by the triplet:

$$e = (k, s_k, j_k)$$

where $k$ is a train, $s_k$ and $j_k$ are respectively the ordinal number of the service and the position of the station in which train $k$ stops. The respective service and station are obtained as follows:

$$s = S(k, s_k) \tag{1}$$

$$j = J(s, j_k) \tag{2}$$

The simulation model updates the events produced in the system dynamically. Fig. 2 shows a real railway network at a certain instant. The example represents three trains with their associated events to be processed.

The event instants $\delta_e$ correspond with the departure from a station, and determine the processing sequence of all the events.

Therefore the basic implementation of the discrete event simulation model is shown in the flow chart depicted in Fig. 3. To process each event it is necessary to consider the two main constraints of the system, the dwell time and the rolling stock. The dwell time constraint is related to a section of the track and models the movement of a train in a track segment. Consider Fig. 4. It shows track segment $i \rightarrow j$ and two trains. Denote by $T^d_{si}$ the departure time of service $s$ performed by train $k$ from station $i$. Denote $T^a_{sj}$ as the instant in which service $s$ arrives at station $j$. The system must satisfy:

$$T^a_{sj} \geq \tau^a_j + d_{ij} \tag{3}$$

$$T^a_{sj} \geq T^d_{si} + c^k_{ij} \tag{4}$$

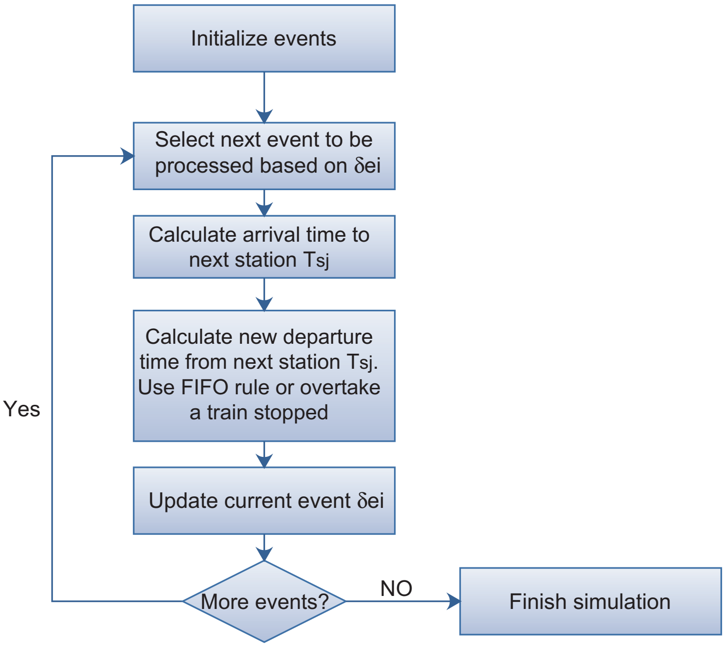
Initialize events
Select next event to be processed based on δei
Calculate arrival time to next station Tsj
Calculate new departure time from next station Tsj. Use FIFO rule or overtake a train stopped
Update current event δei
More events?
Yes
NO
Finish simulation

**Fig. 3.** Discrete event simulation model flow chart.

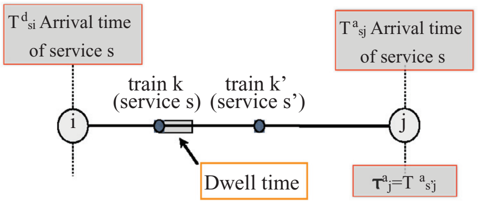

**Fig. 4.** Dwell time constraint.

where $d_{ij}$ is the dwell time of section $i \rightarrow j$, $\tau_j^a$ is the instant when the last train with respect to the previous moment of the system arrives at station $j$ ($k'$ in Fig. 4) and $c_{ij}^k$ is the travel time of the train $k$ in the track segment $i \rightarrow j$.

Eq. (3) imposes a constraint which could lead to a speed reduction for maintaining the minimum security time between trains. Eq. (4) indicates the minimum instant $T_{sj}^a$ in which train $k$ may arrive at station $j$ considering the departure time $T_{si}^d$ from station $i$.

Taking into account both constraints (3) and (4) and that the TTP is focused on an HSR system, the main objective is to reduce the travel time to the minimum. Therefore:

$$T_{sj}^a = \max\left\{T_{si}^d + c_{ij}^k,\ \tau_j^a + d_{ij}\right\} \tag{5}$$

Moreover, it is necessary to consider the stop times of services:

$$T_{sj}^d \geq T_{sj}^a + p_{sj} \tag{6}$$

where $p_{sj}$ is the stop time of service $s$ in station $j$. This approach represents the behaviour of the trains, following a FCFS policy when a train makes a stop in a station.

The model considers an overtaking situation in stations. If a train does not stop in a given station, it could pass the stopped trains in this station. Next, this situation is analysed for service $s$.

Denote by $\tau_j^d$ the programmed departure instant from station $j$ before the arrival of service $s$ at station $j$. The overtaking situation is addressed by calculating:

$$\tau_j^d = \min\left\{\tau_j^d,\ T_{sj}^d\right\} \tag{7}$$

Finally when train $k$ departs from station $j$, the value of $\tau_j^a$ is updated with the value $T_{sj}^a$.

### 3.1.2. Rolling stock constraint

This constraint considers that a planned service for a train cannot start until the previous service of this same train has ended, with a minimum security time and set-up time between them. Suppose there is a train which carries out services $s$ and $s'$ consecutively, then

$$\widetilde{T}_{s'} = \max\{T_{s'},\ \delta_s + r_{ss'}\} \tag{8}$$

where $\widetilde{T}_{s'}$ is the real instant when $s'$ starts, $T_{s'}$ is the planned initial time, $\delta_s$ is the end time of service $s$ and $r_{ss'}$ is the rolling stock turnaround time until the train can start service $s'$.

### 3.1.3. The discrete event algorithm

The discrete event algorithm selects the next event to be processed by determining the train $k$ associated with this event and calculates the arrival time at the next station $j$ given the operational constraints. The arrival of train $k$ at station $j$ updates the last arrival time $\tau_j^a$ at this station, allowing overtaking if the constraints associated with $\tau_j^d$ are met.

The algorithm ends by generating the new event $e^*$ of train $k$, considering the rolling stock constraints and ending the service.

The complete discrete event algorithm can be seen in Table 3.

## 3.2. HSR demand model

The methods of discrete choice analysis have been widely applied to modelling of transportation systems. These methods provide a framework to use the maximum likelihood estimation technique for calibrating and contrasting the parameters of these models. In the multinomial logit and nested logit models, the choice probability can be calculated exactly from the closed-form formula and this has attracted a lot of attention from researchers. Hensher, Rose, and Greene (2005), Louviere, Hensher, and Swait (2000), or Ben-Akiva and Lerman (1995) can be referred to for a more detailed overview of these methods.

**Table 3**
Discrete event simulation algorithm.

```
1. (Parameter initialisation). $\mathcal{E} = \{\emptyset\}$ and $\tau_j^a = -\infty$, $\tau_j^d = -\infty$ for each $j \in \mathcal{J}$.
2. (Event initialisation). For each $k \in \mathcal{K}$, $s_k = 1$ and $j_k = 1$. Add to set $\mathcal{E}$ the events $e = (k, 1, 1)$ for all $k$. Consider
        $T_{sj}^d = T_s$                                                    (9)
        $\delta_e = T_s$                                                    (10)
3. (Simulation). While $\mathcal{E} \neq \{\emptyset\}$
   (a) Calculate next event. Consider
        $\delta_{e^*} = \min_{e \in \mathcal{E}}\{\delta_e\}$              (11)
        $e^* = \operatorname{Arg}\min_{e \in \mathcal{E}}\{\delta_e\}$      (12)
   (b) Consider $e^* = (k, s_k, j_k)$, compute next event for train $k$. Define
       $i = J(k, j_k)$ and $j_k = j_k + 1$. Calculate $j = J(k, j_k)$, there are three possibilities:
     (i) (The service has more stops). If $j_k \neq j_F$ then:
        $T_{sj}^a = \max\{T_{dsi} + c_{ij}^k, \tau_j^a + d_{ij}\}$          (13)
        $T_{sj}^d \geq T_{sj}^a + p_{sj}$                                  (14)
        $\tau_j^d = \min\{\tau_j^d, T_{sj}^d\}$                            (15)
        $\tau_j^a = T_{sj}^a$                                              (16)
       Compute new event $e^* = (k, s_k, j_k)$ and $\delta_e^* = T_{sj}^d$.
     (ii) (Actual service has ended, but there are more services for the train).
     If $j_k = j_F$ and $S(k, s_k + 1) \neq s_F$, then: $j_k = 1, s = S(k, s_k), s' = S(k, s_k + 1), s_k = s_k + 1$,
     and $j = J(s, 1)$. Taking into account the Rolling Stock constraint
        $\widetilde{T}_{s'j} = \max\{T_s', \delta_{e^*} + r_{ss'}\}$      (17)
        Compute new event $e^* = (k, s_k, 1)$ and $\delta_{e^*} = T_{s'j.}$
     (iii) (There are no more events for train k). Delete event $e^*$, $\mathcal{E} = \mathcal{E} - \{e^*\}$.
```

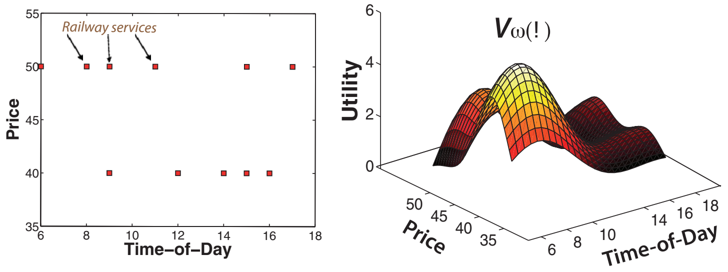

**Fig. 5.** Utillity function.

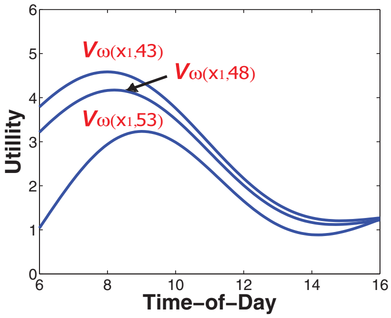

**Fig. 6.** Utillity versus departure time.

Wang, Li, Wang, Moore, Staley, and Li (2014) apply multinomial logit models and nested logit models to analyse traveller mode choice behaviour in a new high-speed rail corridor. In a revealed preference data context, this study reports a high consistency level between the actual and predicted modal splits for business and personal/recreational trips. These models yield reliable estimates and this is a key motivation to use them.

### 3.2.1. *Dynamic utilities modelling*

The utilities for the passengers are calculated following the approach proposed in Espinosa-Aranda et al. (2014). In this subsection we describe this approach and give an illustrative example. Suppose we are analysing a demand pair $\omega = (i, j)$ between station $i$ and $j$. We assume that for the pair $\omega$ there exists a set of rail services $\mathcal{D}_\omega$ that can satisfy this type of journey. We assume that each service is characterized by a set of attributes, and in this example we consider only two: departure time from $i$ and ticket price of the service $s$ to make the journey $\omega$. Fig. 5 shows, on the left, a set $\mathcal{D}_\omega$ of 11 services of which 6 have a cost of 50 euros and the other 5 have a cost of 40 euros. Furthermore, they all have different departure times from station $i$, ranging from 6 in the morning to 18. The attribute vector of service $s$ for demand $\omega$ is $x_{\omega,s} = (b_{\omega,s}, T^d_{s,i})$ where $b_{\omega,\, s}$ is the ticket price and $T^d_{s,i}$ the departure time of service $s$ from station $i$.

We assume that the systematic component of the utility function is given by $V_\omega(x) : X \mapsto \mathbb{R}$, and belonging to a *Reproducing Kernel Hilbert Space* (RKHS). The RKHS are uniquely determined by the function $K(x, y) : X \times X \mapsto \mathbb{R}$ called Mercer kernel. The set of functions $\{K(x, y)\}_{y\in X}$ forms an RKHS basis and thus a utility function $V_\omega$ may be expressed as a linear combination of the elements of the basis

$$V_\omega(x) = \sum_{y\in X} \alpha_y K(x, y), \quad \text{for all } x \in X \tag{18}$$

To make the definition (18) work, we consider a finite number of elements of the basis, specifically those elements associated with services $\mathcal{D}_\omega$, whose corresponding functions are $\{K(x, x_{\omega,s})\}_{s\in\mathcal{D}_\omega}$ and this leads to the following approximation:

$$V_\omega(x) \cong \sum_{s\in\mathcal{D}_\omega} \alpha_{\omega,s} K(x, x_{\omega s}), \quad \text{for all } x \in X \tag{19}$$

Typically (see for instance Ben-Akiva and Lerman (1995)) the systematic utilities are expressed as a linear combination of the attributes, by means of parameters which can be interpreted as the weights that users give to each attribute. In the proposed approach, each utility is expressed as a linear combination of known utilities, given by the parameters $\alpha_{\omega,\, s}$, and the function $K(x, x_{\omega,\, s})$ gauges the similarity between an arbitrary timetable and the known timetables.

Espinosa-Aranda et al. (2014) develop an estimation procedure for these functions (19) based on estimating the utilities $\{V_\omega(x_{\omega,s})\}_{s\in\mathcal{D}_\omega}$ of the set of services involved, and from these estimations calculating the coefficients $\alpha_{\omega,\, s}$ using *Theory of Tikhonov Regularisation*. In this study we assume that the estimation procedure has already been carried out and so the functions $V_\omega(x)$ are known.

This paper uses the Gaussian kernel $K(x, y) = e^{-a\|x-y\|^2}$ in which $\|\cdot\|$ is the Euclidean norm $a \in \mathbb{R}^+$. If we apply the estimation procedure to this example of 11 services we obtain an analytical expression of the form:

$$V_\omega(x_1, x_2) \cong \sum_{s=1}^{11} \alpha_{\omega,s} \exp\Big\{ - a\Big(\big(T^d_{s,i} - x_1\big)^2\Big) + (b_{\omega,s} - x_2)^2\Big\}, \quad \text{for all } x \in X \tag{20}$$

where the coefficients $\alpha_{\omega,\, s}$, $b_{\omega,\, s}$ and $T^d_{s,i}$ are known and fixed. The variables are the attributes $x_1$ and $x_2$ of the service whose utility we wish to calculate. The graph of function (20) of the example is shown on the right of Fig. 5. This expression allows new utilities to be calculated if certain attributes of the service are varied. For example new tariffs may be considered, taking $x_2 = 43, 48$ or $53$ the utility seen as a function of departure time. Fig. 6 shows the three utility functions $V_\omega(x_1, 43)$, $V_\omega(x_1, 48)$ and $V_\omega(x_1, 53)$. This paper uses the functions $V_\omega$ to calculate the utility of the service $\widehat{s}$ when the schedule is modified to $x_1 = T^d_{\omega,\widehat{s}}$.

The formulation of the utility functions considers a single type of user. The definition can be easily broadened to the situation where there are a number of user types $\ell \in \mathcal{L}$ and each group has its own utility function $V^\ell_\omega(x)$. The inclusion of multiple user types in the HSR–TTP model is important not only to be able to consider different sociological characteristics of the population, but also to address one of the key elements of schedule-based dynamic transit assignment for low frequency transit systems (see Wilson & Nuzzolo, 2004), that is, the segmentation of the OD demand flows according to Desired Departure Times (DDT). The functions $V^\ell_\omega$ account for time-varying demand profiles and express when it is most useful for users of type $\ell$ to travel. For example, Fig. 6, shows a user type whose greatest utility is to travel first thing in the morning, but there may be other user types, with their corresponding utility function, where the greatest utility is reached in the afternoon. Each type of user will thus choose their most suitable rail services.

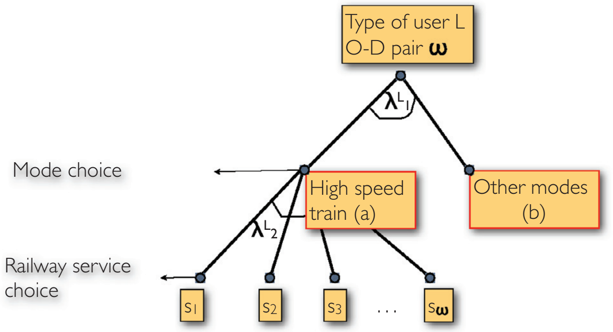

**Fig. 7.** Hierarchical multinomial logit demand for railway service choice.

### 3.2.2. *Constrained nested multinomial logit model*

The demand model used for this HSR is a Constrained Nested Multinomial Logit (CNMNL) model which solves the problem of calculating the behaviour of the passengers. In the HSR–TTP model, the proposed demand model allows us to establish implicitly a relation between the attributes of the timetable and travel behaviour and mode choice. The complete definition and the process by which it was calibrated is described in Espinosa-Aranda et al. (2014).

This model supposes there are various types of users defined by different reasons for travelling, economic and social characteristics and origin-destination (Fig. 7).

The types of users $\ell$ are disaggregated into two components $\ell = (\ell_1, \ell_2)$. The first member includes the socio-economic characteristics and reason for travel; the second component represents the origin and destination of the trip. Each type of user is denoted as $(\ell, \omega)$. Index $\ell \in \mathcal{L}$ refers to the first factors and $\omega = (i, j) \in \mathcal{W}$ is a trip from station $i$ to station $j$.

The model assumes a potential demand $\{\widehat{g}^{\ell}_{\omega}\}_{\omega\in\mathcal{W}}$ for a demand type $\omega$ for users $\ell \in \mathcal{L}$. The total demand is disaggregated in two alternatives

(a) (high-speed) train trips.
(b) Other modes: another means of transport.

Considering this fact, a logit model which divides the potential demand between alternatives ($a$) and ($b$) is defined:

$$g^{m\ell}_{\omega} = \frac{\exp(\lambda^{\ell}_{1} V^{m\ell}_{\omega})}{\sum_{m\in\{a,b\}} \exp(\lambda^{\ell}_{1} V^{m\ell}_{\omega})} \cdot \widehat{g}^{\ell}_{\omega} \qquad m \in \{a, b\}, \ell \in \mathcal{L} \tag{21}$$

where $V^{m\ell}_{\omega}$ is the utility of alternative $m$ for user type $\ell$ and demand type $\omega$.

The model considers a nested logit model for disaggregating demand considering the feasible timetable for a trip type $\omega$. $\mathcal{S}_{\omega}$ represents the feasible set of services for making a trip type $\omega$. The second level of the nested logit model disaggregates the demand between the different services

$$g^{a\ell}_{\omega,s} = \frac{\exp(\lambda^{\ell}_{2} V^{a\ell}_{\omega,s})}{\sum_{s'\in\mathcal{S}_{\omega}} \exp(\lambda^{\ell}_{2} V^{a\ell}_{\omega,s'})} \cdot g^{a\ell}_{\omega} \qquad s \in \mathcal{S}_{\omega}, \ell \in \mathcal{L} \tag{22}$$

Nested logit models calculate the utility of alternative $V^{a\ell}_{\omega}$ as the “log-sum” of the utilities of each service

$$V^{a\ell}_{\omega} = \frac{1}{\lambda^{\ell}_{2}} \ln\left(\sum_{s\in\mathcal{S}_{\omega}} \exp\left(\lambda^{\ell}_{2} V^{a\ell}_{\omega,s}\right)\right) \tag{23}$$

Note that within set $\mathcal{S}_{\omega}$ it is possible to include other HSR services offered by another company, or even air services, to model competition between these means of transport. In this case, these services could not change their planned schedules. Alternative $b$ represents any other alternative not considered in $a$, such as using a private vehicle, not making the journey, going by air (without considering the schedules in disaggregated form), etc.

The above nested logit model is combined with the capacity constraints of the trains. When a train reaches station $j$ the vehicle has picked up passengers from preceding stations. The number of passengers that can take the train is then restricted by the capacity of the vehicle. $\mathcal{W}^{+}_{sj}$ represents the set of origin–destination pairs whose users take the service $s$ before station $j$ and leave the vehicle after station $j$. Also $\mathcal{W}_{sj}$ represents the origin–destination pairs $\omega$ whose origin is station $j$ and use $s$. The capacity constraint of the service $s$ in station $j$ is formulated as:

$$\sum_{\omega'\in\mathcal{W}_{sj}} g^{a}_{\omega,s} + \sum_{\omega\in\mathcal{W}^{+}_{sj}} g^{a}_{\omega,s} \le K_s \quad \text{for all } s \in \mathcal{S}, \qquad j \in J_s; \tag{24}$$

where $K_s$ is the capacity of train $s$, $\mathcal{S}$ is the set of services and $J_s$ represents the set of stations in which the service $s$ will stop.

Eq. (24) is reformulated, disaggregated by users

$$\sum_{\ell\in\mathcal{L}} \left[ \sum_{\omega'\in\mathcal{W}_{sj}} g^{a\ell}_{\omega,s} + \sum_{\omega\in\mathcal{W}^{+}_{sj}} g^{a\ell}_{\omega,s} \right] \le K_s \quad \text{for all } s \in \mathcal{S}, \qquad j \in J_s; \tag{25}$$

The theory of consumer behaviour uses the law of utility maximizing to explain how users choose among their alternatives. Anas (1983) demonstrated that *information minimizing* (or *entropy-maximizing modelling*) and *utility maximizing* should be seen as two equivalent views of the same problem.

CNMNL adopts the information-minimizing approach, which makes possible the addition of constraints such as (25) into its formulation. These constraints introduce new information in the forecasting process in order to represent the complex issue of externalities of the decisions of an individual. This approach leads to the following optimisation problem:

$$\begin{aligned}
\text{minimize} \quad & \textstyle\sum_{\omega\in W} \sum_{\ell\in\mathcal{L}} \Big[ \sum_{m\in\{a,b\}} \eta^{m\ell} g^{m\ell}_{\omega} (\ln g^{m\ell}_{\omega} - 1) + \eta^{c\ell} \\
& \times \textstyle\sum_{s\in\mathcal{S}_w} g^{a\ell}_{\omega,s} (\ln g^{a\ell}_{\omega,s} - 1) - V^{b\ell}_{\omega} g^{b\ell}_{\omega} - \sum_{s\in\mathcal{S}_w} V^{a\ell}_{\omega,s} g^{a\ell}_{\omega,s} \Big] \\
\text{subject to:} \quad & g^{a\ell}_{\omega} + g^{b\ell}_{\omega} = \widehat{g}^{\ell}_{\omega}, \omega \in W, \ell \in \mathcal{L} \\
& g^{a\ell}_{\omega} = \textstyle\sum_{s\in\mathcal{S}_{\omega}} g^{a\ell}_{\omega,s}, \omega \in \mathcal{W}, \ell \in \mathcal{L} \\
& \textstyle\sum_{\ell\in\mathcal{L}} \Big[ \sum_{\omega'\in\mathcal{W}_{sj}} g^{a\ell}_{\omega,s} + \sum_{\omega\in\mathcal{W}^{+}_{sj}} g^{a\ell}_{\omega,s} \Big] \\
& \le K_s, s \in \mathcal{S}, \quad j \in J_s; \quad [\texttt{CNMNL}]
\end{aligned}$$

where $\eta^{a\ell} = \frac{1}{\lambda^{\ell}_{1}} - \frac{1}{\lambda^{\ell}_{2}}$, $\eta^{b\ell} = \frac{1}{\lambda^{\ell}_{1}}$, $\eta^{c\ell} = \frac{1}{\lambda^{\ell}_{2}}$.

CNMNL represents an equilibrium game among rail service users. Users choose the service to make a trip in order to maximize expected utility. Two users who compete for the capacity of the train (there is no capacity for both to travel at once) take part in a form of bidding process. The multipliers of the active capacity restrictions (24) impose (through a mechanism implicit in the optimisation model) a reduction of the utility of full trains. This causes some users (depending on their alternatives) to reject the choice of full trains for other possible alternatives. This mechanism means that in an equilibrium situation the capacity constraints can be satisfied. In other words, there is a nominal utility $V_{\omega,s}$ for each service, and a lower effective utility $\tilde{V}_{\omega,s}$ caused by the capacity constraints, which is the utility experienced by users, and is used by them to take decisions. It should be noted that in this process all users compete against one another to get tickets, and users who board the train first have no advantage over those who board later (we assume on-line systems for ticket sales in which seats are allocated for all journeys).

### 3.3. *Objective function*

This model considers the maximisation of profit, that is, income of ticket sales minus operating costs.

The income of ticket sales is defined as $\sum_{\omega\in\mathcal{W}} \sum_{s\in\mathcal{S}_W} g^{a\ell}_{\omega,s} b_{\omega,s}$ where $g^{a\ell}_{\omega,s}$ is the number of passengers of demand $\omega$ who use the service $s$ and $b_{\omega,s}$ is the price of the ticket for a demand $\omega$ using service $s$.

Let $\pi_s(T_s)$ be the operating costs of service $s$ at instant $T_s$ and $K_s$ the capacity of the train that operates service $s$, we define *the effective capacity* as:

$$\widehat{K}_s(T_s) := \begin{cases} K_s & \text{if } T_s < \mathcal{T} \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

and the *effective operating cost* as:

$$\widehat{\pi}_s(T_s) := \begin{cases} \pi_s(T_s) & \text{if } T_s < \mathcal{T} \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

These definitions represent the fact that if a service $s$ is operated in the planned period $[0, \mathcal{T}]$ then the cost of this service is $\pi_s$ with a capacity $K_s$. Otherwise, the operating cost of this service is 0 and it cannot serve the demand.

Thus the objective function is stated as:

$$Z = \sum_{\ell \in \mathcal{L}} \sum_{\omega \in \mathcal{W}} \sum_{s \in \mathcal{S}_w} g^{a\ell}_{\omega,s} b_{\omega,s} - \sum_{s \in \mathcal{S}} \widehat{\pi}_s(T_s) \tag{28}$$

It is worth noting that if a service $s$ satisfies $T_s > \mathcal{T}$ then it is not profitable to operate this service. If $\mathcal{T} = +\infty$ all the planned services will be scheduled, and the operating costs of the HSR network will be constant with respect to the decision vector $T$. In this case the intention is to align the timetable with the desired trip of each user.

### 3.4. The HSR–TTP timetable model

The optimisation model shown in Fig. 1 represents a bi-level structure. The higher level designs the rail schedules and at the lower level users who are aware of the rail services offered, choose a specific means of transport to make the trip. Conceptually the model can be given as:

$$\begin{array}{lll}
\text{maximize} & Z = \sum_{\ell \in \mathcal{L}} \sum_{\omega \in \mathcal{W}} \sum_{s \in \mathcal{S}_w} g^{a\ell}_{\omega,s} b_{\omega,s} - \sum_{s \in \mathcal{S}} \widehat{\pi}_s(T_s) & \\
\text{subject to:} & \mathbf{T}^d = \texttt{Simulation}(\mathbf{T}) & \texttt{HSR-TTP} \\
& \mathbf{V} = \texttt{v}^*(\mathbf{T}^d) & \\
& \widehat{K}_s = \widehat{K}_s(\widetilde{T}_s) & \\
& \mathbf{g}^a = \texttt{CNMNL}(\mathbf{V}, \widehat{\mathbf{K}}) & \\
& \mathbf{T} \in \mathbb{R}^{|\mathcal{S}|} &
\end{array}$$

The HSR-TTP model has, as decision variables, the vector of departure times of the services $\mathbf{T}$ from the initial stations. This variable can, in principle, take any value. The simulation model calculates the complete timetable $\mathbf{T}^d$ (using Table 3), solving any conflict that might appear and modifying the planned schedules to satisfy the rolling stock constraints. From the schedule $\mathbf{T}^d$ the utilities of the services available are calculated for each journey pair $\omega$. Each service $s$ which lies outside the planning horizon $[0, \mathcal{T}]$ will not, in the end, be run, so its effective capacity will be $\widehat{K}_s = 0$ and the running cost $\widehat{\pi}_s = 0$. Finally, from the effective capacities $\widehat{\mathbf{K}}$ and the utilities $\mathbf{V}$ the demand for the set of services is calculated with the CNMNL model.

This process: (i) running a simulation model to calculate the complete timetable, (ii) evaluating the non-linear utilities and (iii) solving a convex program for estimating the demand disaggregation per service, can be represented mathematically as $g^{a\ell}_{\omega,s}(\mathbf{T})$ to show that there exists an (implicit) function which allows the demand of users of type $\ell$ of service $s$ making journey $\omega$ to be calculated. This allows the model to be formulated as the following unconstrained optimisation problem:

$$\begin{array}{lll}
\text{maximize} & Z(\mathbf{T}) = \sum_{\ell \in \mathcal{L}} \sum_{\omega \in \mathcal{W}} \sum_{s \in \mathcal{S}_w} g^{a\ell}_{\omega,s}(\mathbf{T}) b_{\omega,s} - \sum_{s \in \mathcal{S}} \widehat{\pi}_s(T_s) & \\
\text{subject to:} & \mathbf{T} \in \mathbb{R}^{|\mathcal{S}|} & \texttt{HSR-TTP}
\end{array}$$

## 4. Metaheuristic algorithm

The choice of the solution method should be based on the following three key features of the HSR–TTP timetable model:

- *Unconstrained nonlinear optimisation and derivative-free methods.* The HSR-TTP model is an unconstrained nonlinear optimisation problem. Obtaining closed formulae for the derivatives of the objective function $Z(\mathbf{T})$ is impossible because they are implicitly defined, or they do not exist and so any choice of a resolution method should be made from among the existing unconstrained nonlinear optimisation derivative-free methods.
- *Computational cost.* The computational cost of evaluating the objective function is high because it requires a simulation model to be run and the optimisation model CNMNL to be solved; this should be considered and parallel computing strategies sought.
- *The* HSR-TTP *model presents multiple local optima.*

These characteristics make it impossible to use exact methods and require the use of metaheuristic algorithms. Nowadays there are many algorithm options available (ver Boussaïd, Lepagnot, & Siarry, 2013). This paper considers the Nelder–Mead method (NM, see Nelder and Mead (1965)), Standard Particle Swarm Optimisation (SPSO, see Zambrano-Bigiarini, Clerc, and Rojas (2013)) and a hybridisation of both methods (SPSO+NM, see Espinosa-Aranda, Garcia-Rodenas, and Angulo (2013)). The SPSO+NM algorithm has demonstrated good performance on various unconstrained benchmark functions (see Espinosa-Aranda et al., 2013) and it has also been applied successfully to cluster analysis (Lopez-García, García-Ródenas, & González Gómez, 2014). The HSR-TTP model is computationally demanding and it requires parallelisation strategies to obtain satisfactory solutions in real instances. The SPSO and SPSO+NM algorithms allow these strategies and this is a key motive for its choice. This is a new feature with respect to the implementation used in Espinosa-Aranda et al. (2013).

### 4.1. Standard particle swarm optimisation

*Particle Swarm Optimisation* (PSO, Kennedy and Eberhart (1995)) is a global optimisation technique based on the metaphors of social interaction and communications. PSO has been shown to be effective in solving difficult and complex optimisation problems in a wide range of fields.

The PSO method is probabilistic and maintains a *population* of candidates. This algorithm uses collaboration among a population of simple search agents called *particles* (potential solutions to the problem) to find optima in some search space. The use of population sets helps the algorithm avoid being trapped at a local optimum. In this paper, the Standard PSO 2011[1] is applied, but it includes parallel computing for the particle performance.

In the initialisation phase of the PSO algorithm, the position and velocities of the particles are randomly generated. In every iteration, each particle is updated by following two “best” values. The first, the so-called *pbest*, is achieved by self experience (Step 3) and the second, the so-called *lbest*, through the experience of a subset of particles (Step 4). The set of particles that informs a particle is called its *neighbourhood* . After finding these two values, the position and velocity of each particle will be updated in Step 5.

The complete computational procedure of the Standard PSO algorithm for solving HSR–TTP timetable model is summarized in Table 4.

### 4.2. Nelder–Mead simplex method

The *Nelder–Mead simplex method* (NM) was introduced in Nelder and Mead (1965) and is a technique for minimizing an objective

[1] Several SPSO 2011 programs have been put on line at Particle Swarm Central, URL http://www.particleswarm.info/Programs.htm.

**Table 4**
Standard PSO algorithm for the HSR-TTP timetable model.

Step 1. *(Initialisation)*. Initialize the parameters: number of iterations $(N)$, number of particles $(S)$, learning factor $(c)$, inertia weight $(w)$ and the average number of "informants"' $(K)$. Clerc (2012) suggests the values $S = 40$, $\omega = \frac{1}{2\log(2)} \simeq 0.721$, $c = 1/2 + \log(2) \simeq 1.193$ and $K = 3$.
Initializate population with random positions and velocities.

$$T^0_{p,s} = Rand(0, \mathcal{T}),$$

$$v^0_{p,s} = Rand\left(-T^0_{p,s}, \mathcal{T} - T^0_{p,s}\right),$$

where $Rand(0, \mathcal{T})$ is a random number in $[0, \mathcal{T}]$ drawn according to the uniform distribution. The subindex $p$ is associated with the *pth* particle and the subindex $s$ with the $s$th dimension of the decision variable (departure time of service $s$). Let $\mathcal{N}^0_p$ be the set of neighbours of the particle $p$ at iteration $t = 0$. Let $\widetilde{Z}_p = -\infty$, let $\mathbf{T}^{t-1}_p$ be the $p$th particle at iteration $t - 1$ and set $t = 1$.

Step 2. *(Evaluation [parallel computing])*. Evaluate the objective function (28) for each particle $p$. That is,

$$Z^{t-1}_p = Z\left(\mathbf{T}^{t-1}_p\right) \text{ with } p = 1, \cdots, S. \tag{29}$$

Step 3. *(Updatepbest)*. Find the previous best value for each particle $p$. If the current value $Z^{t-1}_p$ of particle $p$ is better than its optimal value in history, then set the current optimum $\mathbf{B}_p = \mathbf{T}^{t-1}_p$ as the new best value of particle $p$.

Step 4. (Update *lbest*). Find the best position found so far by any particle in the neighbours of the particle $p$ as follows:
Let $p' = \text{Arg} \underset{q \in \mathcal{N}^{t-1}_p}{\text{maximize}} \{Z(\mathbf{B}_q)\}$, if $Z(\mathbf{B}_{p'})$ is greater than the current value $\widetilde{Z}_p$, then $\widetilde{Z}_p = Z(\mathbf{B}_{p'})$ and $\mathbf{L}_p = \mathbf{B}_{p'}$.

Step 5. *(Update the velocity and position)*. If $\mathbf{L}_p \neq \mathbf{B}_p$ then set $c^{t-1}_p = c$; otherwise $c^{t-1}_p = \frac{3c}{4}$. Let $\mathbf{G}^{t-1}_p = \mathbf{T}^{t-1}_p + c^{t-1}_p \cdot (\frac{\mathbf{B}_p + \mathbf{L}_p - 2\mathbf{T}^{t-1}_p}{3})$ and let $\widehat{\mathbf{T}}^{t-1}_p$ be a random point in the hyper-sphere with center $\mathbf{G}^{t-1}_p$ and radius $\|\mathbf{G}^{t-1}_p - \mathbf{T}^{t-1}_p\|$.
Step 5.1. Update the particle's velocity for each particle $p$ according to the following equation

$$v^t_p = w \cdot v^{t-1}_p + \widehat{\mathbf{T}}^{t-1}_p - \mathbf{T}^{t-1}_p \text{ with } p = 1, \cdots, S. \tag{30}$$

Step 5.2. Update the particle's position

$$\mathbf{T}^t_p = \mathbf{T}^{t-1}_p + v^t_p \text{ with } p = 1, \cdots, S. \tag{31}$$

Step 6 *(Stopping criterion)*. If the number of iterations satisfies $t = N$, Stop; otherwise continue. If the current iteration is unsuccessful (no improvement of the best known fitness value), define new neighbourhoods $\mathcal{N}^t_p$ using an adaptative random topology; otherwise $\mathcal{N}^t_p = \mathcal{N}^{t-1}_p$. Let $t = t + 1$ and go back to Step 2.
Exit: the timetable $\mathbf{L}_{p'}$ and its optimal value $Z^* = Z(\mathbf{L}_{p'})$, where $p' = \text{Arg} \underset{p \in \{1, \cdots, S\}}{\text{maximize}} \{Z(\mathbf{L}_p)\}$.

**Table 5**
Nelder–Mead algorithm for HSR-TTP timetable model.

**Step 1.** *(Initialisation)*. For the maximisation of the function $Z(\mathbf{T})$ with $|\mathcal{S}| = n$ variables, create $n + 1$ vertex points to form a $n-$dimensional simplex. Evaluate the objective function at each vertex point of the simplex and sort $(Z(\mathbf{T}_1) \geq Z(\mathbf{T}_2) \geq \ldots \geq Z(\mathbf{T}_{n+1}))$.

**Step 2.** *(Reflection)*.

**Step 2.1.** In each iteration, determine $\mathbf{T}_{low}, \mathbf{T}_{seclow}, \mathbf{T}_{high}$ vertices, indicating vertex points that have the lowest, the second lowest, and the highest function values, respectively. Let $Z_{low}, Z_{seclow}, Z_{high}$ represent the corresponding observed function values.

**Step 2.2.** Find $\mathbf{T}_{cent}$, the centre of the simplex excluding $\mathbf{T}_{low}$ in the maximisation case.

**Step 2.3.** Generate a new vertex $\mathbf{T}_{refl} = (1 + \alpha)\mathbf{T}_{cent} - \alpha\mathbf{T}_{low}$, where $\alpha$ is the reflection coefficient $(\alpha > 0)$ [Nelder and Mead suggested $\alpha = 1$].

**Step 2.4.** If $Z_{high} \geq Z_{refl} \geq Z_{seclow}$, accept the reflection by replacing $\mathbf{T}_{low}$ with $\mathbf{T}_{refl}$, and go back to Step 2.

**Step 3.** *(Expansion)*.

**Step 3.1.** Should reflection produce an objective function greater than $Z_{high}$ (i.e., $Z_{refl} > Z_{high}$), the reflection is expanded in order to extend the search space in the same direction and the expansion point is calculated by: $\mathbf{T}_{exp} = \gamma\mathbf{T}_{refl} + (1 - \gamma)\mathbf{T}_{cent}$, where $\gamma$ is the expansion coefficient $(\gamma > 1)$ [Nelder and Mead suggested $\gamma = 2$].

**Step 3.2.** If $Z_{exp} > Z_{high}$, the expansion is accepted by replacing $\mathbf{T}_{low}$ with $\mathbf{T}_{exp}$; otherwise, $\mathbf{T}_{refl}$ replaces $\mathbf{T}_{low}$. The algorithm continues with a new iteration in Step 2.

**Step 4.** *(Contraction)*.

**Step 4.1.** When $Z_{refl} < Z_{seclow}$ and $Z_{refl} \geq Z_{low}$, then $\mathbf{T}_{refl}$ replaces $\mathbf{T}_{low}$ and contraction is tried.

**Step 4.2.** If $Z_{exp} < Z_{low}$, then direct contraction without the replacement of $\mathbf{T}_{low}$ by $\mathbf{T}_{refl}$ is performed.

**Step 4.3.** The contraction vertex is calculated by: $\mathbf{T}_{cont} = \beta\mathbf{T}_{low} + (1 - \beta)\mathbf{T}_{cent}$, where $\beta$ is the expansion coefficient $(0 < \beta < 1)$ [Nelder and Mead suggested $\beta = 0.5$].

**Step 4.4.** If $Z_{cont} \geq Z_{low}$, the contraction is accepted by replacing $\mathbf{T}_{low}$ with $\mathbf{T}_{cont}$ and then a new iteration begins with Step 2.

**Step 5.** *(Shrink)*.

**Step 5.1.** If $Z_{cont} < Z_{low}$ in Step 4, contraction has failed and shrinkage will be the next attempt. This is done by shrinking the entire simplex by: $\mathbf{T}_i = \delta\mathbf{T}_i + (1 - \delta)\mathbf{T}_{high}$, where $\delta$ is the shrinkage coefficient $(0 < \delta < 1)$ [Nelder and Mead suggested $\delta = 0.5$].

**Step 5.2.** The algorithm then evaluates the objective function at each vertex (except $\mathbf{T}_{high}$) and returns to Step 2 to start a new iteration.

**Exit:** The timetable $\mathbf{T}_{high}$ and its optimal objective function value $Z_{high}$.

function in a many-dimensional space. The NM method is commonly used in unconstrained nonlinear optimisation, which is a well-defined numerical method for twice differentiable and unimodal problems. However, the NM technique is a heuristic search method that may converge to non-stationary points. This method is a direct search method that does not use numerical or analytic derivatives and has very pronounced local convergence. It is therefore an algorithm with no local random component.

The NM method uses the concept of a simplex, which is a special polytope of $n + 1$ vertices in $n$ dimensions. The NM algorithm initializes the procedure with a non-degenerate simplex whose vertices approach the global optimum. This simplex is modified either by changing any of its vertices for another in which the objective function has increased, or going around the polyhedron if it has not improved the objective function. Candidate replacement points are obtained by transforming the worst vertex through reflection, expansion, contraction and shrinkage operations on the centroid of the current simplex.

The main advantage of this method is that it frequently gives significant improvements in the first few iterations and quickly produces quite satisfactory results. This property is very useful in applications where the objective function is too expensive to evaluate Table 5.

### 4.3. A hybridisation of particle and Nelder–Mead simplex method

Metaheuristics often incorporate an exploration (diversification) stage and an exploitation (intensification) stage. Exploration is needed to identify parts of the feasible region with high quality solutions. In the exploitation stage the metaheuristic intensifies the search in some promising areas previously detected.

Espinosa-Aranda et al. (2013) propose a framework to hybridize two metaheuristics by using each in the exploration and exploitation stages. In this paper, we apply an instance of this algorithmic class, called SPSO + NM. The SPSO + NM algorithm applies the SPSO

**Table 6**
SPSO+NM algorithm for HSR–TTP timetable model.

| | |
|---|---|
| Step 1. | (*Initialisation*). Initialize the number of iterations ($N$), the SPSO parameters and randomly generate an initial population of particles. Initialize the parameters $n_c$, $n_r$ associated with the number of iterations carried out by SPSO and NM methods respectively. Set the counters $t = 1$ and $t_{\text{SPSO}} = 0$ and let $\widehat{Z} = -\infty$. |
| Step 2. | (*Exploration stage*). Apply one iteration of SPSO algorithm to the current population. Let $\mathbf{L}_{p'}$ be the current best solution and $Z(\mathbf{L}_{p'})$ its objective value. If $\widehat{Z} > Z(\mathbf{L}_{p'})$, then let $t_{\text{SPSO}} = t_{\text{SPSO}} + 1$ and $\widehat{Z} = Z(\mathbf{L}_{p'})$. |
| Step 3 | (*Exploitation stage*). If $t_{\text{SPSO}} = n_c$ apply $n_r$ iterations of NM algorithm by initializing the method from $\mathbf{L}_{p'}$. Let $\mathbf{T}_{high}$ be the solution found, then set $\mathbf{L}_{p'} = \mathbf{T}_{high}$ and take $t_{\text{SPSO}} = 0$. Update $\widehat{Z} = Z(\mathbf{T}_{high})$. |
| Step 4 | (*Stopping criterion*). If the current iteration is $t = N$, Stop; otherwise set $t = t + 1$ and go to Step 2.<br>Exit: The timetable $\mathbf{L}_{p'}$ and its objective function $Z(\mathbf{L}_{p'})$. |

**Table 7**
Types of railway service on Madrid–Seville corridor.

| Type of service | Route | Amount of services | Type of train |
|---|---|---|---|
| 1 | MAD → CR → PU | 11 | AVANT |
| 2 | MAD → CR → PU → COR | 3 | AVE |
| 3 | MAD → CR → PU → COR → SEV | 5 | AVE |
| 4 | MAD → COR | 4 | AVE |
| 5 | MAD → COR → SEV | 9 | AVE |
| 6 | MAD → SEV | 3 | AVE |
| 7 | COR → SEV | 6 | MD |
| 8 | COR → SEV | 9 | AVANT |
| 9 | SEV → COR | 6 | MD |
| 10 | SEV → COR | 9 | AVANT |
| 11 | SEV → COR → PU → CR → MAD | 5 | AVE |
| 12 | SEV → COR → MAD | 8 | AVE |
| 13 | SEV → MAD | 3 | AVE |
| 14 | COR → PU → CR → MAD | 3 | AVE |
| 15 | COR → MAD | 5 | AVE |
| 16 | PU → MAD → CR | 11 | AVANT |

**Table 8**
Features of each service on Madrid–Seville corridor.

| Type | Average speed (kilometers/hour) | Capacity ($K_s$) | Operating cost/kilometer (euros) |
|---|---|---|---|
| AVE | 215 | 308 | 17.5 |
| AVANT | 193 | 237 | 16.4 |
| MD | 106 | 190 | 10.2 |

**Table 9**
Potential total trips ($\hat{g}_w$).

| | MA | CR | PU | COR | SEV |
|---|---|---|---|---|---|
| **MA** | – | 1377 | 479 | 185 | 3270 |
| **CR** | 1377 | – | 216 | 185 | 185 |
| **PU** | 479 | 216 | – | 185 | 185 |
| **COR** | 185 | 185 | 185 | – | 1291 |
| **SEV** | 3270 | 185 | 185 | 1291 | – |

method until $n_c$ improvements of the objective function are obtained (exploration phase). Then the NM algorithm performs $n_r$ iterations starting from the best solution found by the SPSO algorithm (exploitation phase). Finally, the best solution in the population set is replaced with the solution found in the exploitation stage and the process is repeated. The parameters $n_c$ and $n_r$ play the role of providing a trade-off between the processes of global and local search, i.e. between exploration stage and exploration stage.

Espinosa-Aranda et al. (2013) show that SPSO+NM achieves a faster convergence than SPSO. This is because SPSO+NM detects that the search neighbourhood has changed and so applies a local search based on NM. This search is systematic, with no random component, unlike the original SPSO, which accelerates the process.

The SPSO+NM algorithm applied to the HSR-TTP timetable model is shown in Table 6.

## 5. Computational experiments

This section tries to test the proposed methodology based on derivative free methods focusing on two main objectives: (i) studying the quality of the solution found and (ii) studying the computational efficiency of the algorithms.

This model has been implemented in a software application called *dhAVE* (which can be seen running in http://youtu.be/K9mAoo__5Uk), and run in MATLAB 2013, while the demand model has been solved using CONOPT. The computer used to perform the computational experiments has the following characteristics: Windows 7 64 bits, processor: 2 × AMD Opteron 4226, 12 cores 2.7 gigahertz, RAM: 12 gigibyte 1600 megahertz.

### 5.1. Case study

To test the HSR–TTP model, a case study has been generated. This numerical example is focused on the Madrid–Seville corridor of the HSR network of Spain. This corridor is formed by 5 stations [Madrid(MAD), Ciudad Real(CR), Puertollano(PU), Cordoba(COR), Seville(SEV)] which produces 20 origin-destination demand pairs (10 per direction of travel) serving 15115 passengers/day. Currently this demand is completely covered by 100 services. The data used for the experiment can be downloaded from http://bit.ly/1gCFw5e. Fig. 8 depicts the corridor used by these services, Table 7 indicates the route of each type of service and Table 8 shows the characteristics of each type of train.

The origin–destination matrix (different potential trips between stations) on a working day is shown in Table 9.

The baseline in order to compare the timetable obtained consists of the current schedule used by RENFE after running the current planned schedule in the simulation module. The simulated time-station diagram is shown in Fig. 9. To compare the solution obtained with the actual schedule of RENFE avoiding distortions, the operation of all services has been forced, that is $\mathcal{T} = +\infty$.

The demand model computes the utility for each train and each demand. In this example only one type of user ($\ell = 1$) is considered with regard to economic characteristics, but taking into account the trip of each traveller (origin–destination pair $\omega$), there will be 20 types of user. Each user type $\omega$ can travel using a set of services $s \in S_\omega$. Considering the planned schedule, the set of alternatives $(\omega, s) \in \mathcal{A}$ consists of 298 alternatives.

In this computational experiment we have considered the utility functions based on Gaussian kernels estimated in Espinosa-Aranda et al. (2014). The attributes considered for each possible alternative $(\omega,\ s)$ are: (i) $T^d_{sj}$, the departure time from station $j \in \mathcal{J}$ of service $s$, (ii) $b_{\omega,s}$, the price for demand $\omega$ using service $s$ and (iii) $c^s_w$, the travel time of service $s$ for demand $w$ and . The vector of attributes is denoted as $x_{\omega s} = (T^d_{sj}, b_{\omega,s}, c^s_w)$. To simplify, Espinosa-Aranda et al. (2014) set $V_\omega(x) = V^*(x)$ for all pairs $\omega$. So the utility $V^{a\ell}_{\omega,s}$ for each type of users $\ell$ (in this case, $\ell = 1$) using service $s$ for the demand $\omega$ is calculated using Eq. (19) as follows:

$$V^{a\ell}_{\omega,s} = V^*\left(T^d_{sj}, b_{\omega,s}, c^s_w\right) \tag{32}$$

It is worth noting that the value of $T^d_{sj}$ is calculated by the optimisation model and the attributes $b_{\omega,s}$, $c^s_w$ are fixed.

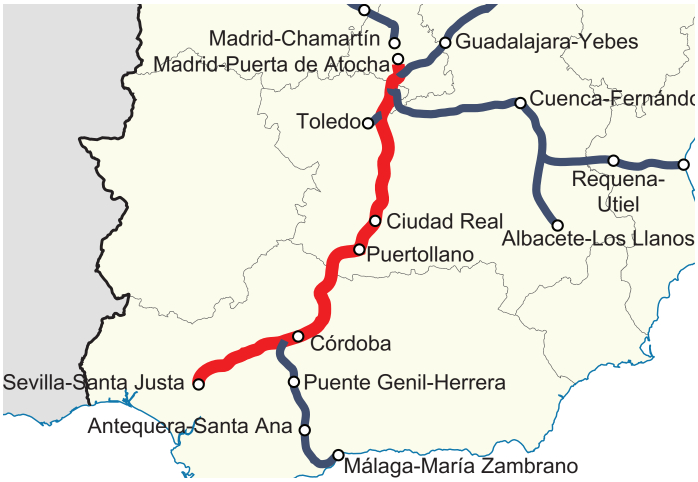

**Fig. 8.** Madrid–Seville corridor.

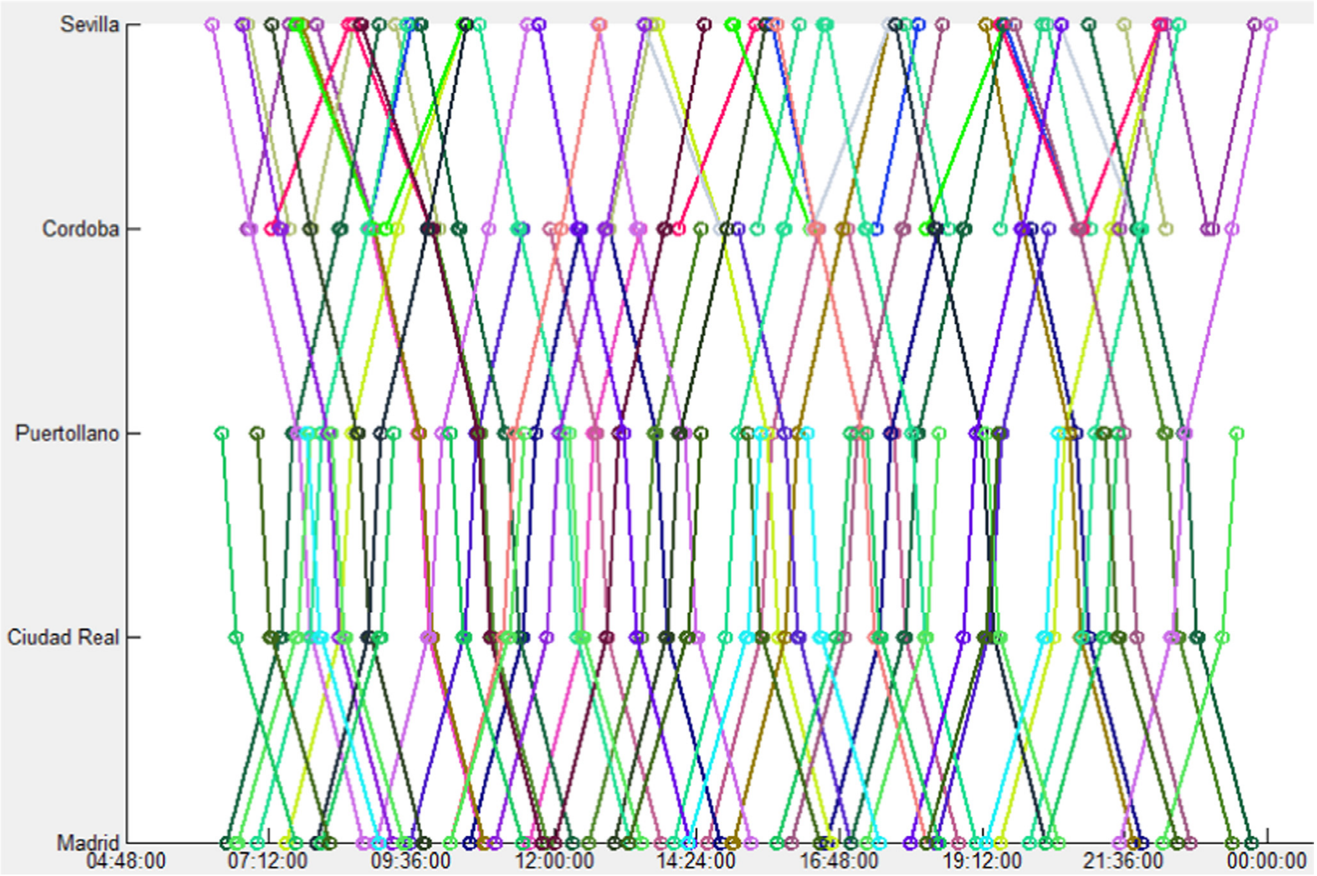

**Fig. 9.** Time–station graph obtained by simulation.

The baseline value is calculated using the current RENFE schedule with an objective value of 3, 64, 180 euros/day.

### 5.2. Experiment I: qualitative analysis of solution

This section shows a set of 11 experiments using the NM, SPSO and SPSO+NM algorithms. Each algorithm has been run for 10, 000 objective function evaluations. The NM algorithm starts from the planned schedule to obtain its proposed solution. SPSO and SPSO+NM methods start with an initial swarm of 40 particles calculated randomly on the interval [0:00, 23:59] hours. The parameters $w$, $c_1$ and $c_2$ for updating velocity are defined as $w = 1/(2\ln(2))$, $c_1 = 0.5 + \ln(2)$ and $c_2 = c_1$ (see Zambrano-Bigiarini et al., 2013). SPSO+NM values $n_c$ and $n_r$ are set to 1, 5, 25 and 1, 5, 10 percent of the total iterations respectively, combining all the possibilities.

The results obtained are shown in Table 10. The first column indicates the algorithm used and its configuration. Also the solution found and the CPU time of the algorithm can be shown in the second and third columns respectively. The last column contains the improvement obtained with respect to the baseline value using the

**Table 10**
Improvements obtained by NM, SPSO and SPSO+NM algorithms.

| Algorithm | Solution found (euros) | CPU time (hours) | Improvement (percent) |
|---|---|---|---|
| NM | 367836 | 6.3 | 1.00 |
| SPSO | 371642 | 6.1 | 2.04 |
| SPSO+NM | | | |
| $(n_c = 1, n_r = 1)$ | 373130 | 6.1 | 2.45 |
| $(n_c = 1, n_r = 5)$ | 372833 | 6.6 | 2.37 |
| $(n_c = 1, n_r = 10)$ | 371904 | 7.5 | 2.12 |
| $(n_c = 5, n_r = 1)$ | 372032 | 6.1 | 2.15 |
| $(n_c = 5, n_r = 5)$ | 372344 | 6.6 | 2.24 |
| $(n_c = 5, n_r = 10)$ | 372816 | 7.0 | 2.37 |
| $(n_c = 25, n_r = 1)$ | 372709 | 6.1 | 2.34 |
| $(n_c = 25, n_r = 5)$ | 372228 | 6.6 | 2.20 |
| $(n_c = 25, n_r = 10)$ | 373160 | 7.2 | 2.46 |

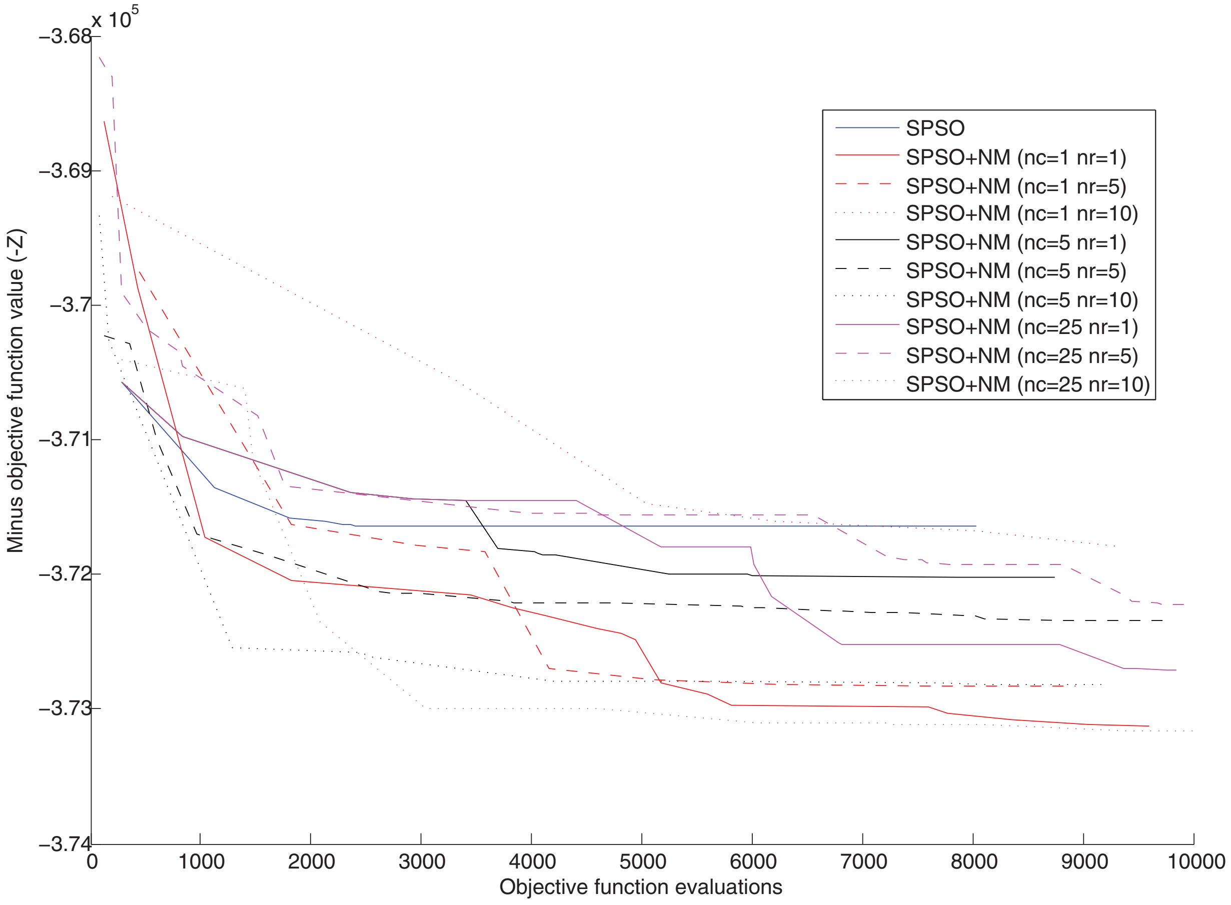

**Fig. 10.** Performance of algorithms.

following equation:

$$\texttt{Improvement} = \frac{Z - \text{baseline value}}{\text{baseline value}} * 100 \qquad (33)$$

As can be seen, the computational time of the optimisation algorithms is between 6.1 and 7.5 hours producing an improvement percentage around 1 percent in the NM case and more than 2 percent in SPSO-related algorithms. It is worth noting that the SPSO+NM approach obtained in all cases a better solution than NM and SPSO algorithms alone.

Moreover, Fig. 10 shows the objective value versus the number of evaluations of the objective function.

### 5.3. Experiment II: test of parallelisation strategies

This section analyses the possibility of reducing the computational time of the proposed optimisation algorithms. SPSO and SPSO+NM algorithms can be improved because parts of them can be parallelized. There is software code that can be changed from sequential (see Fig. 11) to parallel mode (see Fig. 12). This part then changes its behaviour and several evaluations of the objective function can be done at the same time.

Using this idea, the speed up-ratio is obtained for each algorithm using the following equation:

$$\texttt{speedup} - \texttt{ratio}_n = \frac{\text{sequential runtime}}{\text{parallel runtime with } n \text{ cores}} \qquad (34)$$

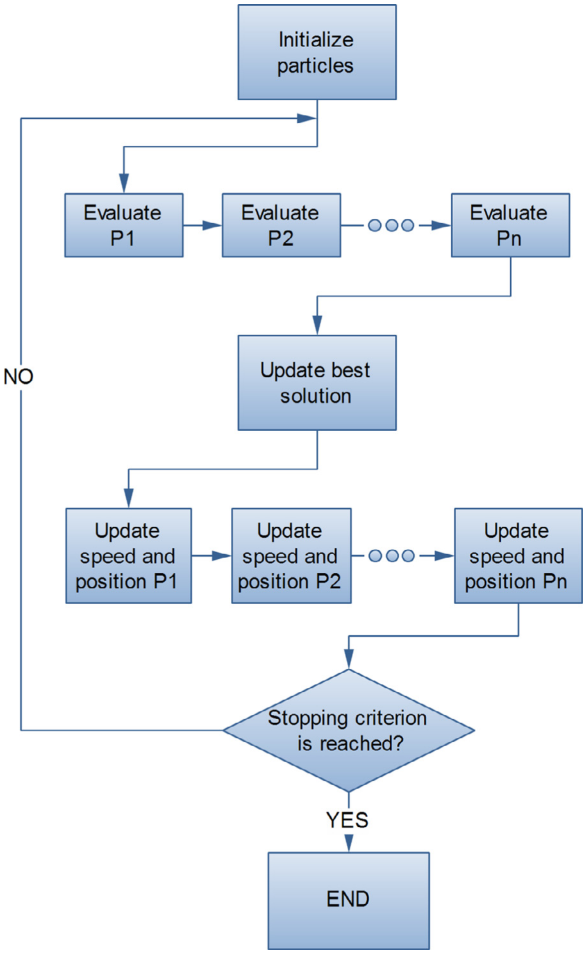
Initialize particles
Evaluate P1
Evaluate P2
Evaluate Pn
NO
Update best solution
Update speed and position P1
Update speed and position P2
Update speed and position Pn
Stopping criterion is reached?
YES
END

**Fig. 11.** Sequential SPSO.

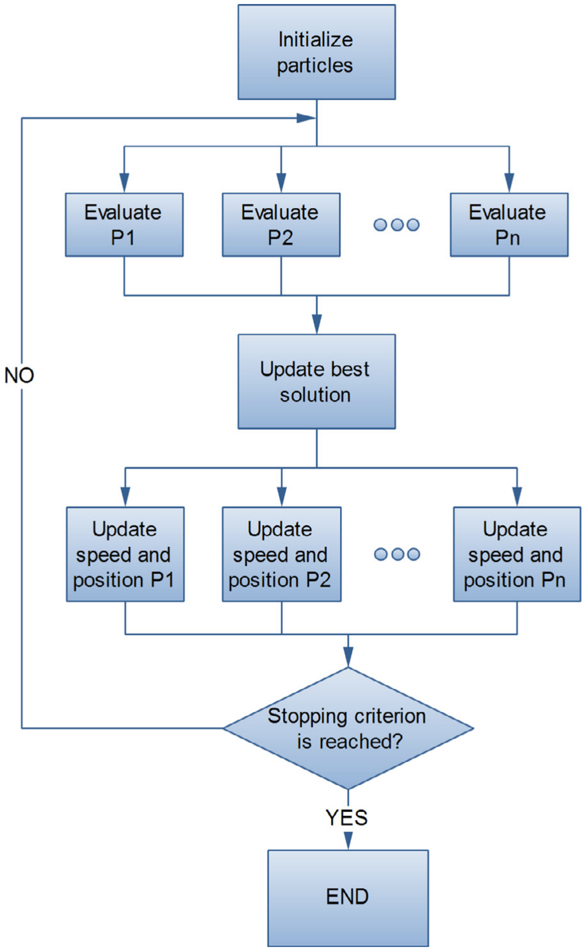
Initialize particles
Evaluate P1
Evaluate P2
Evaluate Pn
NO
Update best solution
Update speed and position P1
Update speed and position P2
Update speed and position Pn
Stopping criterion is reached?
YES
END

**Fig. 12.** Parallel SPSO.

**Table 11**
Computational comparison between parallel algorithms.

| Algorithm | CPU time (hours) | Speed up-ratio |
|---|---|---|
| SPSO | | |
| 2 cores | 3.5 | 1.74 |
| 4 cores | 1.8 | 3.39 |
| 8 cores | 1.1 | 5.54 |
| 12 cores | 1.0 | 6.10 |
| SPSO+NM $n_c = 25,\ n_r = 10$ | | |
| 2 cores | 3.7 | 1.94 |
| 4 cores | 1.8 | 4.00 |
| 8 cores | 2.1 | 3.43 |
| 12 cores | 2.0 | 3.60 |

This experiment has been carried out running the SPSO and SPSO+NM ($n_c = 25,\ n_r = 10$) algorithms in parallel mode with 2, 4, 8 and 12 parallel threads. This configuration of the algorithm has been selected because it is the one which found the better solution. The results of the computational cost and `speed-up ratio` of the algorithms can be seen in Table 11 and Fig. 13.

The results show that the improvement of each algorithm depends on the number of parallel threads. However it is worth noting that the `speed up-ratio` in the SPSO+NM algorithm does not increase with the number of parallel threads. This occurs because of the random nature of the algorithm, in which the NM algorithm, which is non-parallelizable, was run an indeterminate number of times depending on the solutions found by the SPSO stage. The case with 2 and 4 parallel threads only ran the NM algorithm once, performing 1000 function evaluations, while with 8 and 12 cores the NM algorithm is run twice, performing 2000 function evaluations.

In conclusion, the best algorithm is the SPSO+NM hybridisation which achieves an improvement of 2.46 percent and the parallelisation strategies are capable of reducing the computational time of this algorithm to 2 hours.

### *5.4. Experiment III: scalability of the resolution methods*

The goal of this subsection is to analyse the impact of problem size and characteristics on computing times and solution quality. In this experiment we have modified the size of the instances by adding/removing trains from the timetable baseline, and we have solved these examples with the NM, SPSO and SPSO+NM($n_c = 25,\ n_r = 10$) methods, using the same set of parameters as in the previous experiments. We have also considered the parallelized version with 12 cores of the SPSO and the SPSO+NM.

Fig. 14 shows the CPU times. The significant facts are that, on the one hand the SPSO in its sequential version tends not to be applicable in big problems, while the parallel version, having a higher performance as shown in Experiment *II*, is the one that behaves best with respect to computational cost.

However, since metaheuristics do not provide upper bounds, to assess solution quality, Fig. 15 shows a comparison between the solutions. It can be seen that all the methods except NM converge to solutions with the same value in the objective function. The sole exception is the NM which appears to converge to a worse solution when the size of the problem grows. These results may be explained

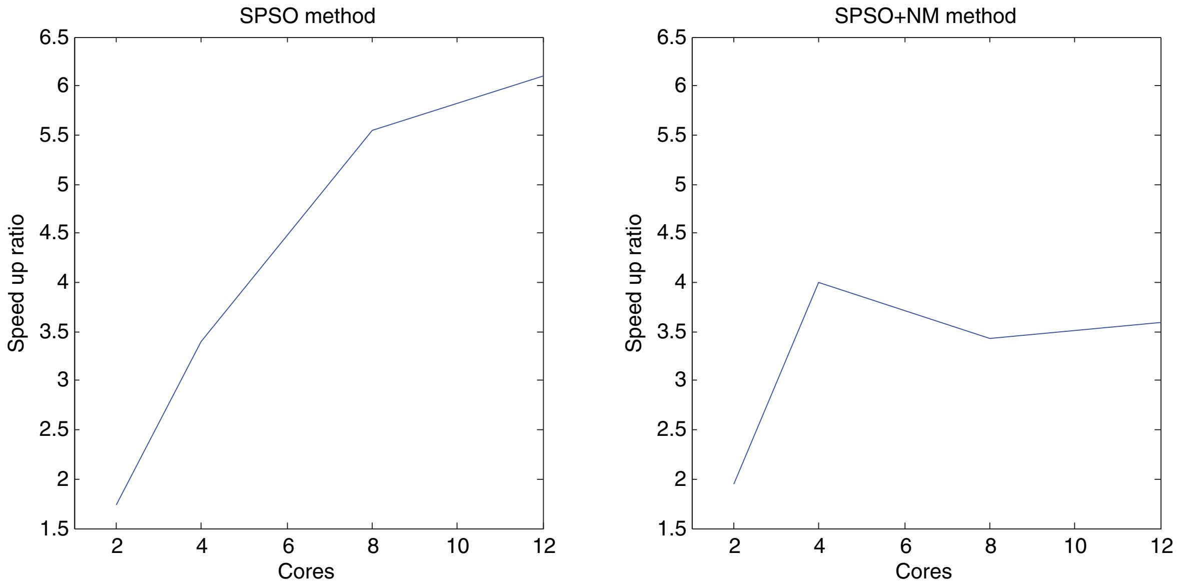

**Fig. 13.** `Speed-up ratio` for SPSO and SPSO+NM algorithms.

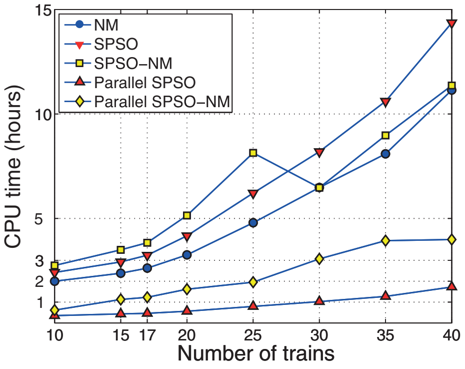

**Fig. 14.** Performance of algorithms.

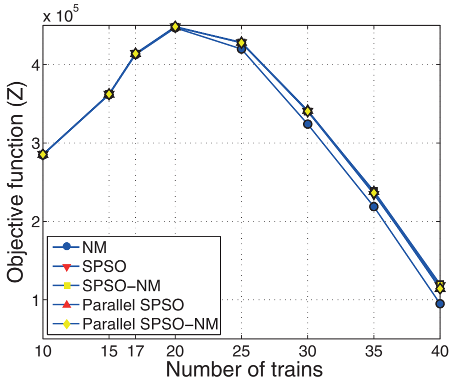

**Fig. 15.** Quality of solutions.

because the algorithms based on SPSO methods are global optimisation methods whereas the NM has local convergence.

## 6. Conclusion

This paper proposes a disaggregated *multi-modal* schedule-based HSR model focused on endogenous factors, to determine the HSR demand. This model is divided into two modules. The first is a discrete event simulation model which calculates the whole timetable considering the departure time of the trains from the origin station and several operational constraints. After this, a novel constrained nested logit model obtains the number of passengers that will use the services based on the calculated departure times, the travel times, the ticket price and the capacity constraints of each service.

The proposed HRS–TTP model aims at maximizing the profits of the railway operator. It is solved using derivative free algorithms, hybridisation techniques and parallelisation techniques in a computational experiment based on real data from the Madrid–Seville High-Speed corridor.

The results show the applicability of the proposed approach to a real scenario in an acceptable computational time and how the model generated represents the behaviour of the users of the railway network considering the timetable calculated, achieving an improvement in the benefits obtained by the railway operator of 2.46 percent with respect to the original schedule. The numerical experiments show also how parallelisation strategies may become essential when addressing large scale instances of the `HSR-TTP` model.

## Acknowledgements

This research has been financed by the *Ministerio de Economía y Competitividad* of Spain with the TRA-2011-27791-C03-03 research project.

## References

Almodóvar, M., & García-Ródenas, R. (2013). On-line reschedule optimization for passenger railways in case of emergencies. *Computers & Operations Research, 40*(3), 725–736.

Anas, A. (1983). Discrete choice theory, information theory and the multinomial logit and gravity models. *Transportation Research Part B, 17*(1), 13–23.

Ben-Akiva, M., Cascetta, E., Coppola, P., Papola, A., & Velardi, V. (2010). High speed rail demand forecasting in a competitive market: the italian case study. In *Proceedings of the world conference of transportation research (WCTR), Lisbon, Portugal.*

Ben-Akiva, M., & Lerman, S. (1995). *Discrete choice analysis: Theory and applications to travel demand*. Cambridge, MA: MIT Press.

Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information Science, 237*, 82–117. doi:10.1016/j.ins.2013.02.041.

Börjesson, M. (2014). Forecasting demand for high speed rail. *Transportation Research Part A: Policy and Practice, 70*, 81–92. http://dx.doi.org/10.1016/j.tra.2014.10.010.

Cacchiani, V., & Toth, P. (2012). Nominal and robust train timetabling problems. *European Journal of Operational Research, 219*(3), 727–737.

Cascetta, E., & Coppola, P. (2012). An elastic demand schedule-based multimodal assignment model for the simulation of high speed rail (HSR) systems. *EURO Journal on Transportation and Logistics, 1*(1-2), 3–27.

Cascetta, E., & Coppola, P. (2014). High speed rail (HSR) induced demand models. *Procedia - Social and Behavioral Sciences, 111*, 147–156. http://dx.doi.org/10.1016/j.sbspro.2014.01.047.

Clerc, M. (2012). Standard Particle Swarm Optimisation. <hal-00764996>. https://hal.archives-ouvertes.fr/hal-00764996.

Cordeau, J.-F., Toth, P., & Vigo, D. (1998). A survey of optimization models for train routing and scheduling. *Transportation Science, 32*, 380–404.

Corman, F., D'Ariano, A., Pacciarelli, D., & Pranzo, M. (2010). A tabu search algorithm for rerouting trains during rail operations. *Transportation Research Part B: Methodological, 44*(1), 175–192.

Couto, A., & Graham, D. J. (2008). The impact of high-speed technology on railway demand. *Transportation, 35*(1), 111–128.

D'Ariano, A. (2008). *Improving Real–Time Train Dispatching: Models, algorithms and applications*. (Ph.D. thesis), The Netherlands TRAIL Research School. TRAIL Thesis Series no. T2008/6.

D'Ariano, A., Pacciarelli, D., & Pranzo, M. (2008). Assessment of flexible timetables in real-time traffic management of a railway bottleneck. *Transportation Research Part C, 16*, 232–245.

Espinosa-Aranda, J., & García-Ródenas, R. (2013). A demand-based weighted train delay approach for rescheduling railway networks in real time. *Journal of Rail Transport Planning and Management, 3*(1–2), 1–13.

Espinosa-Aranda, J. L., & García-Ródenas, R. (2012). A discrete event-based simulation model for real-time traffic management in railways. *Journal of Intelligent Transportation Systems, 16*(2), 94–107.

Espinosa-Aranda, J. L., Garcia-Rodenas, R., & Angulo, E. (2013). A framework for derivative free algorithm hybridization. In *Adaptive and natural computing algorithms: vol. 7824* (pp. 80–89). Springer.

Espinosa-Aranda, J. L., García-Ródenas, R., López-García, M. L. Angulo-Herrera, E. (2014). Constrained logit model: Formulation and calibration, Submitted.

Fröidh, O. (2008). Perspectives for a future high-speed train in the swedish domestic travel market. *Journal of Transportation Geography, 16*(4), 268–277.

Fu, X., Oum, T. H., & Yan, J. (2014). An analysis of travel demand in Japan's intercity market empirical estimation and policy simulation. *Journal of Transport Economics and Policy, 48*(1), 97–113.

Hensher, D., Rose, J., & Greene, W. (2005). *Applied choice analysis*. Cambridge: Cambridge University Press.

Hsu, C.-I., & Chung, W.-M. (1997). A model for market share distribution between high-speed and conventional rail services in a transportation corridor. *The Annals of Regional Science, 31*(2), 121–153.

Jiang, X., Zhang, L., & Chen, X. M. (2014). Short-term forecasting of high-speed rail demand: A hybrid approach combining ensemble empirical mode decomposition and gray support vector machine with real-world applications in China. *Transportation Research Part C: Emerging Technologies, 44*, 110–127. http://dx.doi.org/10.1016/j.trc.2014.03.016.

Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. *IEEE International Conference on Neural Networks - Conference Proceedings, 4*, 1942–1948.

Lopez-García, M.L., García-Ródenas, R., & González Gómez, A. (2014). *Hybrid metaheuristic optimization algorithms for time-domain-constrained data clustering*. Technical Report, Universidad de Castilla-La Mancha.

Louviere, J., Hensher, D., & Swait, J. (2000). *Stated choice methods: Analysis and application*. Cambridge: Cambridge University Press.

Mandel, B., Gaudry, M., & Rothengatter, W. (1997). A disaggregate box–cox logit mode choice model of intercity passenger travel in germany and its implications for high-speed rail demand forecasts. *The Annals of Regional Science, 31*(2), 99–120.

Martín, J. C., & Nombela, G. (2007). Microeconomic impacts of investments in high speed trains in spain. *The Annals of Regional Science, 41*(3), 715–733.

Nelder, J., & Mead, R. (1965). A simplex method for function minimization . *Computer Journal, 7*(4), 308–313.

Park, Y., & Ha, H.-K. (2006). Analysis of the impact of high-speed railroad service on air transport demand. *Transportation Research Part E: Logistics and Transportation Review, 42*(2), 95–104.

Román, C., Espino, R., & Martín, J. C. (2007). Competition of high-speed train with air transport: The case of Madrid–Barcelona. *Journal of Air Transport Management, 13*(5), 277–284.

Sánchez-Rico, M. T., García-Ródenas, R., & Espinosa-Aranda, J. L. (2014). A monte carlo approach to simulate the stochastic demand in a continuous dynamic traffic network loading problem . *Intelligent Transportation Systems, IEEE Transactions, 15*(3), 1362–1373.

Wang, Y., Li, L., Wang, L., Moore, A., Staley, S., & Li, Z. (2014). Modeling traveler mode choice behavior of a new high-speed rail corridor in China. *Transportation Planning and Technology, 37*(5), 466–483. doi:10.1080/03081060.2014.912420.

Wardman, M. (2006). Demand for rail travel and the effects of external factors. *Transportation Research Part E: Logistics and Transportation Review, 42*(3), 129–148.

Wilson, N. H., & Nuzzolo, A. (2004). *Schedule-based dynamic transit modeling: Theory and applications (operations research/computer science interfaces, 28)*. Norwell, MA, USA: Kluwer Academic Publishers.

Yao, E., & Morikawa, T. (2005). A study of on integrated intercity travel demand model. *Transportation Research Part A: Policy and Practice, 39*(4), 367–381.

Zambrano-Bigiarini, M., Clerc, M., & Rojas, R. (2013). Standard particle swarm optimisation 2011 at cec-2013: A baseline for future PSO improvements. In *IEEE congress on evolutionary computation (CEC), 2013* (pp. 2337–2344). IEEE.