03_IF_10_2016 pag_769_788__ 02/11/16 09:05 Pagina 769

Il sistema informatico "Treno Blu"
Motore di ricerca per viaggi ferroviari di persone a ridotta mobilità

The "Blue Train" computer system
Search engine for rail travel of people with reduced mobility

Dott. Ing. Alberto FRANCIOSI(*)
Dott. Ing. Ilaria PETROSELLI(**)

1. Premessa

Attualmente sono disponibili vari sistemi informatici di consultazione degli orari ferroviari, utili per la programmazione di itinerari di viaggio.

Tra questi:

– il Quadro Orario di Rete Ferroviaria Italiana (RFI) [A];

– i motori orari delle Imprese ferroviarie disponibili sui loro siti web e su applicazioni per *smartphone* [B].

Tali sistemi sono basati su algoritmi di ricerca impostati in funzione delle finalità che sottendono (acquisto di biglietti, ricerca di soluzioni ottimali, …). Di conseguenza non sempre permettono la ricerca di soluzioni di viaggio particolari in maniera semplice e completa.

I Quadri Orari di RFI (fig. 1), ad esempio, consentono solamente di verificare quali treni partono o arrivano in una certa fascia oraria nella stazione di riferimento. Quindi il passeggero che deve raggiungere una certa stazione di destino, non direttamente collegata a quella di partenza, deve costruire autonomamente tutti i percorsi possibili.

L'Impresa ferroviaria Trenitalia, invece, mette a disposizione un software applicativo online in grado di individuare, indicate la stazione di partenza, quella di arrivo e la fascia oraria d'interesse, tutte le possibili soluzioni di viaggio ovviamente con servizi di Trenitalia. Le soluzioni di viaggio che vengono fornite includono tutti i percorsi disponibili, anche

1. Introduction

Currently we offer a variety of useful train timetable enquiry computer systems for scheduling travel itineraries.

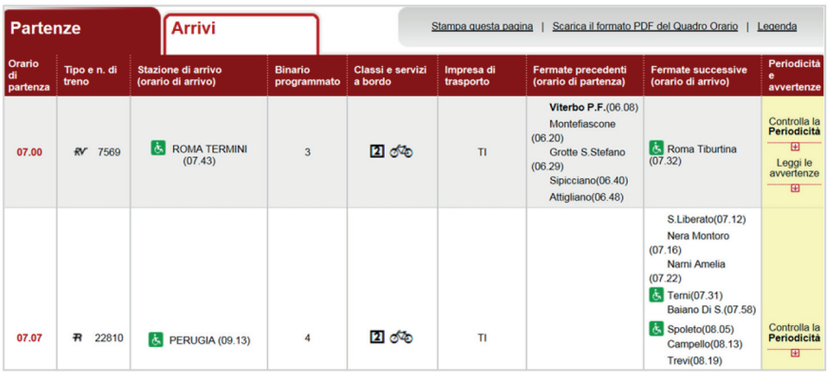
Among these:

– *the Timetable of Rete Ferroviaria Italiana (RFI) [A];*

– *railway Undertakings time engines available on their websites and on applications for* smartphones [B].

These systems are based on search algorithms that are set according to the underlying purposes (purchase of tickets, search for optimal solutions, etc.). Therefore they do not always allow searching for particular travel solutions in a simple and complete way.

RFI Timetables (fig. 1) for example, allow only to see which trains leave or arrive at a certain time slot in the ref-



Partenze | Arrivi | Stampa questa pagina | Scarica il formato PDF del Quadro Orario | Legenda

Orario di partenza	Tipo e n. di treno	Stazione di arrivo (orario di arrivo)	Binario programmato	Classi e servizi a bordo	Impresa di trasporto	Fermate precedenti (orario di partenza)	Fermate successive (orario di arrivo)	Periodicità e avvertenze
07.00	RV 7569	ROMA TERMINI (07.43)	3	2	TI	Viterbo P.F.(06.08) Montefiascone (06.20) Grotte S.Stefano (06.29) Sipicciano(06.40) Attigliano(06.48)	Roma Tiburtina (07.32)	Controlla la Periodicità Leggi le avvertenze
07.07	R 22810	PERUGIA (09.13)	4	2	TI		S.Liberato(07.12) Nera Montoro (07.16) Narni Amelia (07.22) Terni(07.31) Baiano Di S.(07.58) Spoleto(08.05) Campello(08.13) Trevi(08.19)	Controlla la Periodicità

Fig. 1 - Quadro orario online RFI: esempio di soluzioni per treni in partenza da Orte.

Fig. 1 - RFI online timetable: example of solutions for trains departing from Orte.

(*) "Servizi alla Clientela" di "Servizi per la Circolazione" di Direzione Produzione RFI.

(**) Il presente lavoro è stato oggetto della tesi di laurea della Dott.ssa Ing. Ilaria PETROSELLI presso l'Università degli Studi "Sapienza" - Ingegneria Gestionale.

(*) *"Customer Services" of "Traffic Services" by RFI Production Management.*

(**) *This work was the subject of the Dr. Ing. Ilaria* PETROSELLI *dissertation at the "Sapienza" University - Engineering Management.*

03_IF_10_2016 pag_769_788__ 02/11/16 09:06 Pagina 770

con cambio, per spostarsi da una stazione di origine ad una di destinazione con informazioni sui servizi offerti (fig. 2).

Per le persone a ridotta mobilità (PRM) o con disabilità, la ricerca di soluzioni di viaggio accessibili, in quanto caratterizzate dalla idoneità del treno al trasporto e dalla possibilità di servizi di assistenza nelle stazioni, risulta molto laboriosa con i sistemi attualmente esistenti.

I sistemi informativi offerti ai passeggeri dalle imprese ferroviarie infatti presentano due forti limitazioni. Innanzitutto forniscono informazioni esclusivamente sui propri treni, inoltre le persone a ridotta mobilità non hanno sempre la possibilità di utilizzare tutte le soluzioni individuate dal software, in quanto:

– i treni inclusi nel percorso suggerito dal sistema potrebbero non essere attrezzati per il trasporto passeggeri disabili con sedia a ruote;
– le stazioni ferroviarie dove avviene il cambio del treno potrebbero non essere munite degli opportuni servizi necessari per garantire la mobilità del disabile a terra;
– un percorso potrebbe prevedere un cambio treno da svolgersi in un tempo non idoneo ad una PRM.

Allo scopo di offrire un supporto ai passeggeri con disabilità che usufruiscono del treno e alla stessa società RFI, che, oltre l'attività di Gestore dell'Infrastruttura, svolge anche il ruolo di Station Manager e cura l'assistenza ai clienti a ridotta mobilità per garantire loro la completa accessibilità al sistema ferroviario, è stato realizzato un software applicativo che aiuti nella ricerca di soluzioni di viaggio [1].

Il software applicativo, denominato "Treno Blu", analizza i treni di tutte le Imprese ferroviarie, considera solo le stazioni attrezzate alla mobilità dei disabili, pone come vincolo per le connessioni un tempo di interscambio tra treni superiore ai 15 minuti e, se necessario, considera

erence station, hence the passenger who has to reach a certain destination station, not directly connected to the departure one, has to build all possible paths on its own.

The railway company Trenitalia, instead, offers online application software capable of locating all possible travel solutions with Trenitalia services of course, after indicating the departure station, the arrival station and the time slot of interest. Travel options that are provided include all available paths, also with change, to move from an origin station to a destination station with information on the services provided (fig. 2).

For people with reduced mobility (PRM) or with disabilities, search for accessible travel solutions, characterised by the transport ability of the train and by the possibility of assistance in stations, is very hard-working with existing systems.

Information systems offered to passengers by railway undertakings in fact have two severe limitations. Firstly they provide information exclusively on their trains, moreover persons with reduced mobility do not always have the possibility to use all the solutions identified by the software, because:

– *trains included in the route suggested by the system may not be equipped to transport disabled passengers with wheelchairs;*
– *train stations where the change of train takes place may not be equipped with appropriate services needed to ensure mobility of the disabled on the ground;*
– *a route may foresee a train change to be carried out in a time not suitable for a PRM.*

In order to offer support to passengers with disabilities who use the train and to the same company RFI, which, besides the activity of Infrastructure Manager, also plays the role of Station Manager and takes care of customers with reduced mobility to ensure their unimpeded access to

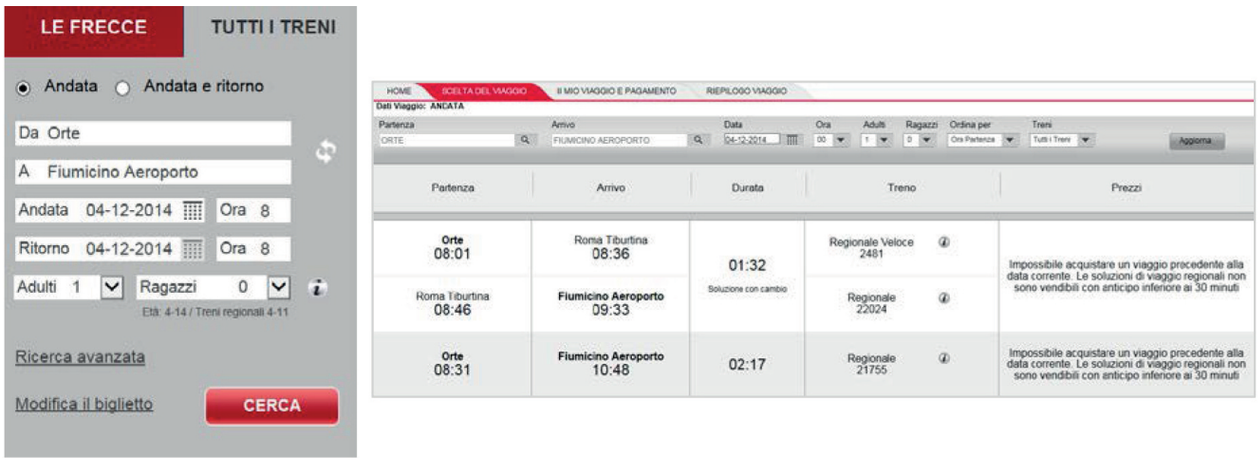


Fig. 2 - Soluzioni di viaggio sito Trenitalia: esempio inserimento stazione di arrivo, partenza e fascia oraria d'interesse.
Fig. 2 - Trenitalia web site travel solutions: example of input of arrival, departure station and time slot of interest.

solo treni attrezzati per il trasporto passeggeri su sedia a ruote.

2. Modellizzazione del problema

Il punto di partenza per la realizzazione del programma è stata la formalizzazione del problema attraverso la costruzione di un adeguato modello matematico che prende in considerazione tutte le caratteristiche essenziali del fenomeno reale in esame [2] [3].

I problemi di ricerca di una soluzione di viaggio tra due stazioni sono riconducibili a problemi di cammino minimo su un grafo orientato[1] opportunamente definito (fig. 3).

In termini matematici è possibile esprimere la ricerca di un cammino minimo dal nodo *u* al nodo *v* attraverso un modello di programmazione lineare intera in cui la funzione obiettivo da minimizzare è la seguente:

$$min \sum_{(i,j)\,\epsilon\,E} w_{ij} \cdot x_{ij}$$

dove:

- x_{ij} è una variabile binaria che assume valore 1 se l'arco (i, j) appartiene ad un cammino da *u* a *v* e 0 altrimenti;
- w_{ij} è il peso dell'arco (i, j).

Affinché un insieme di archi del grafo G costituisca un cammino da *u* a *v* bisogna assicurarsi che un arco del cammino esca da *u* e nessuno entri in esso (condizione 1), che un arco del cammino entri in *v* e nessuno esca da esso (condizione 2) e che per tutti gli altri vertici il numero di archi del cammino uscenti sia pari a quello di archi entranti (condizione 3). Formalizzando si ottengono i seguenti vincoli [4] [5]:

$$\sum_{(i,h)\,\epsilon\,E} x_{hi} - \sum_{(j,i)\epsilon\,E} x_{ij} = \begin{cases} 1, & h = u \quad (1) \\ -1, & h = v \quad (2) \\ 0, & h \in V\backslash\{u,v\} \quad (3) \end{cases}$$

Nella modellizzazione dell'orario ferroviario, sono identificati come nodi, le stazioni del circuito di assi-

[1] Un grafo orientato è una struttura matematica costituito da un insieme di elementi G(V,E) dove V è l'insieme dei nodi (anche chiamati vertici) ed E è l'insieme di archi orientati con funzione costo w: E→R che associa ad ogni arco un peso w a valore nei reali.

Un cammino tra due nodi u e v del grafo è una sequenza di nodi v1, v2,…, vk, tali che u = v1, v = vk ed e = (vi-1, vi) con i = 1,…,k; il cui peso è la somma dei pesi degli archi che lo costituiscono. Un cammino tra una coppia di vertici v1 e vk è detto minimo se non esiste in G un cammino alternativo di costo totale inferiore a w(p).

the railway system, a software application that helps in the search for travel solutions was created [1].

The application software, named "Blue Train", analyses the trains of all railway Undertakings, considers only the stations equipped for mobility of the disabled, poses an interchange time between trains exceeding 15 minutes as a constraint for connections and, if necessary, considers only trains equipped to transport passengers in wheelchairs.

2. Modelling of the problem

The starting point for the implementation of the programme was the formalisation of the problem by constructing a suitable mathematical model that takes into account all the essential features of the real phenomenon under consideration [2] [3].

The problems of finding a travel solution between two stations are related to shortest way problems on an oriented graph[1] appropriately defined (fig. 3). In mathematical terms it is possible to express the search for the shortest path from node u to node v through an entire linear programming model in which the objective function to be minimised is the following:

$$min \sum_{(i,j)\,\epsilon\,E} w_{ij} \cdot x_{ij}$$

where:

- *x_{ij} is a binary variable which takes value 1 if the arc (i, j) belongs to a path from u to v and 0 alternatively;*
- *w_{ij} is the weight of the arc (i, j).*

In order for a set of arcs of graph G to constitute a path from u to v, we must make sure that an arc of the path exists u and none enters it (condition 1), that an arc of the path enters v and none exits it (condition 2) and that for all other vertices the number of outgoing arcs of the path is equal to that of incoming arcs (condition 3). Formalising the following constraints are obtained [4] [5]:

$$\sum_{(i,h)\,\epsilon\,E} x_{hi} - \sum_{(j,i)\epsilon\,E} x_{ij} = \begin{cases} 1, & h = u \quad (1) \\ -1, & h = v \quad (2) \\ 0, & h \in V\backslash\{u,v\} \quad (3) \end{cases}$$

[1] An oriented graph is a mathematical structure consisting of a set of elements G (V, E) where V is the set of nodes (also called vertices) and E is the set of arcs oriented with cost function w: E → R which associates to each arc a weight w in value in real.

A path between two u and v nodes of the graph is a sequence of nodes v1, v2, ..., vk, such that u = v1, v = vk and e = (vi-1, vi) with i = 1, ..., k; whose weight is the sum of the weights of its constituent arcs. A path between a pair of vertices v1 and vk is called minimum if no alternate path of total cost less than w(p) exists in G.

03_IF_10_2016 pag_769_788__ 02/11/16 09:06 Pagina 772

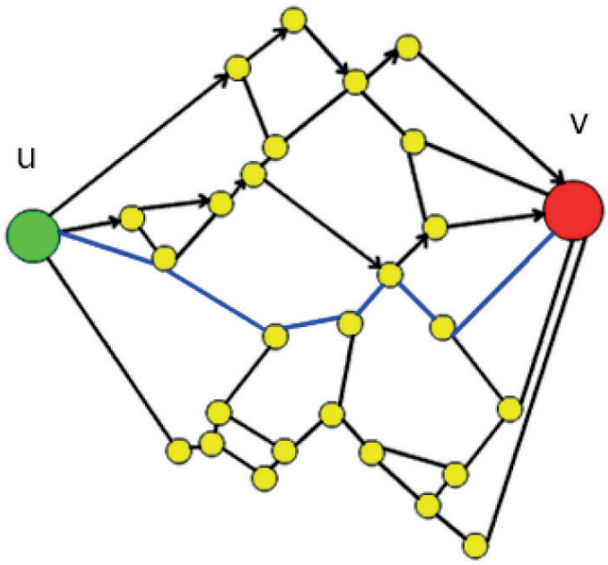


Fig. 3 - Cammino minimo dal nodo u al nodo v.
Fig. 3 - Shortest path from node u to node v.

stenza PRM, mentre come archi, i treni che transitano da una stazione all'altra, per cui si hanno collegamenti dinamici tra i diversi nodi in cui la variabile tempo è di fondamentale importanza, esistono in letteratura due approcci equivalenti dal punto di vista prestazionale che consentono di modellare in maniera opportuna il problema di cammino minimo su grafi dinamici: l'approccio Tempo – Espanso e l'approccio Tempo Dipendente. Si è preferito il primo approccio in quanto la possibilità di superamento tra treni con medesima origine e destinazione, necessaria per la compresenza di reti tradizionali ed ad alta velocità, non è accettabile nel secondo approccio.

L'approccio Tempo – Espanso [C] trasforma un grafo dinamico in uno statico in cui ogni nodo corrisponde ad un determinato evento, ovvero la partenza o l'arrivo di un treno in una determinata stazione, mentre l'arco tra i nodi rappresenta il collegamento tra i due eventi (fig. 4).

Per considerare nel modello i tempi necessari al cambio treno, i nodi raggruppati per stazione sono stati suddivisi in due sottoinsiemi che rappresentano le partenze e gli arrivi ed è stato necessario introdurre un ulteriore sottoinsieme, copia del sottoinsieme dei nodi partenza, definito "nodi di trasferimento".

Per simulare invece ogni tipologia di azione presente nel modello di viaggio in treno è stato necessario individuare 6 categorie di archi (fig. 5).

Arco treno: connette l'istante di partenza di un treno da una stazione con l'istante di arrivo dello stesso treno presso la stazione successiva in cui effettua una fermata, rappresenta cioè lo spostamento in treno tra due stazioni. In base alla periodicità effettiva del treno considerato ed

By modelling train timetables, the PRM assistance circuit stations are identified as nodes, whereas trains transiting from one station to the other are identified as arcs, therefore dynamic links between different nodes in which the time variable is crucial are obtained. There are two equivalent approaches in literature from the performance point of view that allow modelling properly the minimum path problem on dynamic graphs: the Time – Expanded approach and the Time – Dependent approach. The first approach was preferred because the possibility to overtake between trains with same origin and destination, necessary for the coexistence of conventional and high-speed networks, is not acceptable in the second approach.

The approach Time – Expanded [C] changes a dynamic graph into a static one where each node corresponds to a certain event, namely the departure or arrival of a train in a particular station, while the arc between nodes represents the link between the two events (fig. 4).

In order to consider the time required to change train in the model, nodes grouped per station were divided into two subsets representing departures and arrivals and it was necessary to introduce a further subset, copy of the departure nodes subset, called "transfer nodes".

To simulate instead each type of action present in the train journey model it was necessary to identify 6 categories of arcs (fig. 5).

Train arc: *it connects the train departure moment from a station with the same train arrival time at the next station where it stops; it therefore represents the train displacement between two stations. Based on the actual peri-*

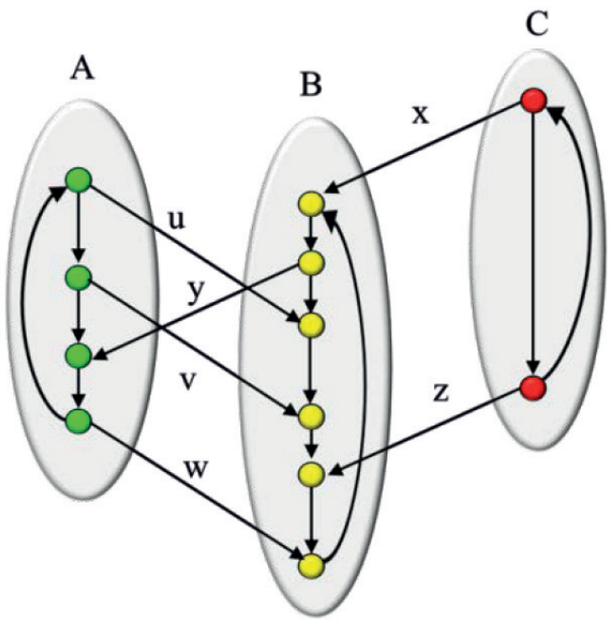


Fig. 4 - Grafo Tempo - Espanso con tre stazioni A, B, C.
Fig. 4 - Time - Expanded graph with three stations A, B, C.

03_IF_10_2016 pag_769_788__ 02/11/16 09:06 Pagina 773

al giorno selezionato per il viaggio, questa connessione elementare può essere attiva oppure no.

Arco attesa: rappresenta il tempo che intercorre tra partenze consecutive di due treni nella stessa stazione.

Arco bordo: indica la possibilità per il passeggero di rimanere a bordo del treno nei minuti che intercorrono tra l'arrivo e la partenza dello stesso treno dalla stessa stazione.

Arco cambio: rappresenta l'azione di discesa da un treno e collega cioè il nodo di arrivo di un treno con il nodo di trasferimento nella stessa stazione successivo all'istante di arrivo di una quantità maggiore o uguale al tempo minimo di trasferimento impostato.

Arco salita: rappresenta l'azione di salita sul treno.

Arco a piedi: rappresenta la possibilità di spostarsi a piedi tra i marciapiedi di una stazione posizionati in punti non ravvicinati tra loro (ad esempio i binari esterni della stazione Tiburtina).

Ad ogni arco è stato associato un peso specifico in base al criterio di ottimizzazione scelto dall'utente per ricercare le soluzioni di viaggio. Nel software è impostata una ricerca multicriterio, in cui l'utente può scegliere tra tre diverse opzioni di ricerca:

– il percorso più rapido;
– il percorso con meno cambi;
– il percorso che sfavorisce i treni ad AV.

Scegliendo una delle tre opzioni, viene attribuita maggiore importanza al criterio selezionato e resi meno importanti gli altri. In altri termini, il peso complessivo dell'arco diventa la somma ponderata dei singoli pesi secondo la seguente formula:

$$c_1 \cdot x + c_2 \cdot y + c_3 \cdot z$$

dove:

– x rappresenta il tempo di percorrenza della tratta in minuti;
– y assume valore maggiore di 0 solo per gli archi di tipo cambio; in particolare 1 se il cambio avviene presso una stazione che richiede 1 ora di preavviso per il servizio di assistenza e 2 nel caso in cui è richiesto un preavviso di 12 ore;
– z è una variabile booleana che assume valori 0 e 1, rispettivamente nel caso in cui il treno sia o non sia ad alta velocità;
– c_1, c_2 e c_3 rappresentano i coefficienti dei singoli criteri, i quali variano a seconda dell'opzione selezionata.

odicity of the train considered and the day chosen for the journey, this elementary connection may be active or not.

Wait arc: *it represents the time between successive departures of two trains in the same station.*

Aboard arc: *it indicates the possibility for passengers to remain aboard the train in the minutes between the arrival and departure of the same train from the same station.*

Change arc: *it represents the action of descent from a train, that is it connects the arrival node of a train with the transfer node in the same station following the moment of arrival of a quantity greater than or equal to the minimum transfer time set.*

Boarding arc: *it represents the boarding action on the train.*

On foot arc: *it represents the possibility to walk between station platforms positioned in points not close between them (for example the external tracks of the Tiburtina station).*

Each arc has been associated with a specific weight depending on the optimisation criterion chosen by the user to search travel options. A multi-criteria research is set in the software, where the user can choose from three different search options:

– *the fastest route;*
– *the route with less changes;*
– *the route that disadvantages HS trains.*

By choosing one of three options, greater importance is attributed to the criterion selected and others are made less important. In other words, the total weight of the arc becomes the weighted sum of the individual weights according to the following formula:

$$c_1 \cdot x + c_2 \cdot y + c_3 \cdot z$$

where:

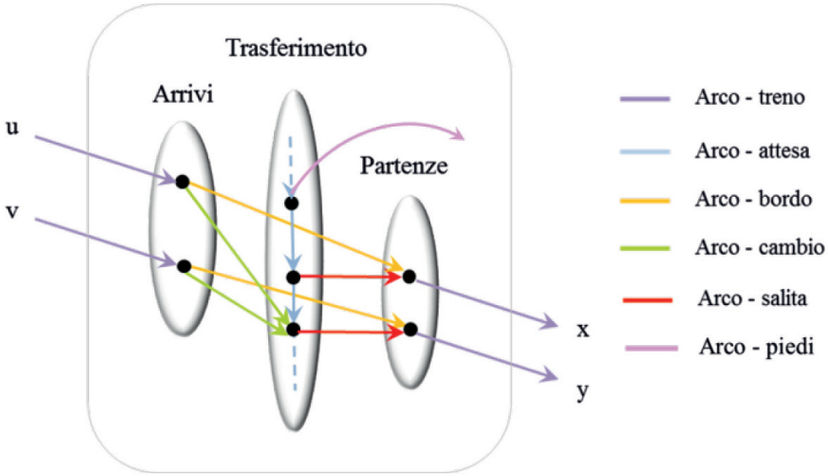


Fig. 5 - Modello di trasferimento nell'approccio Tempo – Espanso.
Fig. 5 - Transfer model in the Time – Expanded approach.

03_IF_10_2016 pag_769_788__ 02/11/16 09:06 Pagina 774

Nello specifico per l'opzione "percorso più rapido" vengono ricercati i percorsi più veloci e, a parità di tempo di percorrenza, vengono selezionati quelli con il minor numero di trasbordi. Per l'opzione "percorso con meno cambi" vengono prima individuati i percorsi con il minor numero di cambi, favorendo le stazioni in cui è richiesto un preavviso di un'ora per il servizio di assistenza, poi a parità di trasbordi viene scelto il percorso più rapido senza considerazioni anche in questo caso sull'economicità del percorso. Per l'opzione "percorso no Alta Velocità" si individuano i percorsi con il minor numero di trasbordi, e a parità di cambi, si scelgono i percorsi più rapidi, sfavorendo i treni Eurostar attraverso una maggiore pesatura degli archi legati ai Treni ad Alta Velocità.

3. Algoritmo di ricerca

L'algoritmo scelto per la ricerca del cammino minimo è stato quello di Dijkstra [D], in quanto più performante in problemi con sorgente singola su un grafo orientato in cui tutti i pesi degli archi siano non negativi.

L'algoritmo di Dijkstra in ogni istante divide l'insieme V dei nodi del grafo in due parti: l'insieme S dei nodi visitati, ovvero i nodi per cui il peso di cammino minimo dalla sorgente s è già stato determinato e l'insieme Q dei nodi sconosciuti ancora da esaminare.

Per ogni nodo v ε Q, l'algoritmo tiene traccia di un valore d[v] che memorizza ad ogni iterazione il peso del cammino migliore trovato sino a quel punto e di una variabile π[v] che memorizza il nodo che precede v. L'array π rappresenta l'output principale dell'algoritmo in quanto da questo è possibile riscostruire a ritroso il cammino minimo verso ogni nodo.

Il principio di funzionamento dell'algoritmo di Dijkstra è definito dal codice riportato in fig. 6.

È prevista una fase di inizializzazione in cui, per ogni nodo v del grafo, la distanza d[v] dalla sorgente viene inizializzata ad infinito ed il predecessore assume valore NULL, mentre al nodo sorgente s si assegna una distanza rispetto a sé pari a 0. L'insieme S dei nodi visitati è definito come un insieme vuoto, mentre l'insieme Q dei nodi sconosciuti è caratterizzato da tutti i nodi del grafo. Ad ogni ciclo "while" si estrae dell'insieme Q, il nodo u a cui corrisponde il valore d[u] minimo, e viene automaticamente inserito nell'insieme S dei nodi esaminati. Un volta individuato il nodo u da estrarre, si costruisce l'insieme frontiera F del nodo, ossia l'insieme di tutti i nodi adiacenti ad u, con lo scopo di verificare se il cammino che porta ad essi passando per u è migliore dei cammini trovati precedentemente. A tal proposito ad ogni arco uscente da u ed entrante in v di peso w si applica la cosiddetta tecnica del rilassamento, che consiste nell'aggiornare la stima d[v] e il predecessore π[v], se il cammino minimo migliora passando per il nodo u. In tal caso d[v] viene definito come la somma tra d[u] e il peso dell'arco che congiunge i due nodi ed il predecessore del no-

– *x represents the travel time of the route in minutes;*
– *y assumes a value greater than 0 only for change type arcs, in particular 1 if the change takes place at a station that requires a 1 hour notice for assistance and 2 in case a 12-hour notice is required;*
– *z is a Boolean variable that takes values 0 and 1 respectively in the event that the train is or is not a high-speed train;*
– *c_1, c_2 and c_3 represent the coefficients of individual criteria, which vary depending on the chosen option.*

Specifically for the "fastest route" option the fastest routes are searched and, with the same travel time, those with the least number of transfers are chosen. For the "route with fewer changes" option the routes with the fewest changes are first identified, favouring stations where one-hour notice is required for assistance, then with equal transfers the fastest route is chosen without consideration in this case of the inexpensiveness of the route. For the "no High-Speed route" option, routes with the least number of transfers are identified, and with the same changes, the fastest routes are chosen, thus disadvantaging Eurostar trains through greater weighing of arcs linked to High-Speed Trains.

3. Search algorithm

The chosen algorithm for searching the shortest path was Dijkstra [D] as it is more efficient in problems with single source on an oriented graph where all weights of the arcs are non-negative.

Dijkstra's algorithm in each instant divides set V of the nodes of the graph into two parts: set S of the nodes visited, namely nodes for which the weight of the shortest path from the source s has already been determined and set Q of unknown nodes still to be examined.

For each v ε Q node, the algorithm keeps track of a d[v] value which stores the weight of the best path found up to that point at each iteration and a variable π[v] which stores the node that precedes v. The π array is the main output of the algorithm because from this it is possible to reconstruct the shortest path towards each node.

The operating principle of Dijkstra's algorithm is defined by the code shown in fig. 6.

An initialisation phase is envisaged where, for each node v of the graph, the distance d[v] from the source is initialised to infinity and the predecessor assumes NULL value, while the source node s is assigned a distance with respect to itself equal to 0. Set S of nodes visited is defined as an empty set, while set Q of unknown nodes is characterised by all nodes in the graph. At each "while" cycle set Q is extracted, node u that has the minimum d[u] value, and is automatically placed in set S of the nodes examined. When node u to be extracted is identified, the F border set of the node is built, i.e. the set of all nodes that are adjacent to u, in order to verify if the path leading to them via u is the best of the paths previously found. In this re-

03_IF_10_2016 pag_769_788__ 02/11/16 09:06 Pagina 775


```
Dijkstra (G, w, s)
Inizialize_Single_Source (G, s)
    for ogni v ϵ V
        d[v] ← ∞
        π[v] ← NULL
    d[s] ← 0
S ? 0
Q ? V[G]
while Q ? Ø
    do u ? EXTRACT-MIN(Q)
        S ? S ∪ {u}
    for ogni vertice v ϵ F[u]
    Relax (u, v, w)
        if d[v]  > d[u] + w (u, v)
            d[v] ← d[u] + w (u, v)
            π[v] ← u
```

Fig. 6 - Pseudocodice algoritmo di Dijkstra.
Fig. 6 - Dijkstra's algorithm pseudo-code.

do v diventa il nodo u. La ricerca termina quando l'insieme Q è vuoto.

Sono possibili diverse implementazioni dell'algoritmo di Dijkstra che differiscono per il modo in cui è analizzato l'insieme Q per la selezione del nodo u con valore d[u] minimo. Queste non cambiano il comportamento dell'algoritmo, ma ne influenzano l'efficienza. L'insieme Q può essere modellato come una coda di priorità, ossia una struttura dati astratta che permette di rappresentare un insieme di elementi, in questo caso nodi, su cui è definita una relazione d'ordine [E].

Esistono tre metodi di implementazione delle coda di priorità:

– *lista non ordinata,* adatta se applicata a grafi densi e fortemente connessi per cui $|E| = |V|^2$,
– *heap di Fibonacci,* ottima per entrambe le situazioni ma molto onerosa, e
– *heap binario,* adatto per grafi sparsi in cui $|E| = |V|$.

Per il sistema in esame in cui il numero di nodi è confrontabile con quello degli archi, la struttura dati più idonea per la gestione dell'insieme Q è l'heap binario (fig. 7).

gard, to each arc outgoing from u and entering v with w weight the so-called relaxation technique is applied, which consists in updating the estimate d[v] and the predecessor π[v], if the shortest path improves passing through node u. In this case d[v] is defined as the sum of d[u] and the weight of the arc which joins the two nodes and the predecessor of node v becomes node u. The search ends when set Q is empty.

There are different implementations of Dijkstra's algorithm that differ in how set Q is analysed for the selection of node u with d[u] minimum value. These do not change the behaviour of the algorithm, but affect its efficiency. Set Q can be modelled as a priority queue, which is an abstract data structure that allows representing a set of items, in this case nodes, on which an order relationship is defined [E].

There are three methods for implementing priority queues:

– unordered list, *suitable if applied to dense and strongly connected graphs for which* $|E| = |V|^2$,
– Fibonacci heap, *great for both situations but very burdensome, and*
– binary heap, *suitable for scattered graphs where* $|E| = |V|$.

The binary heap is the most suitable data structure for managing the Q set for the system where the number of nodes is comparable with that of the arcs (fig. 7).

A binary heap is an almost complete binary tree, that is to say a non-oriented graph that has the following characteristics:

– *two vertices are connected to one path;*
– *the initial node is called root;*
– *each node except the root has only one parent;*

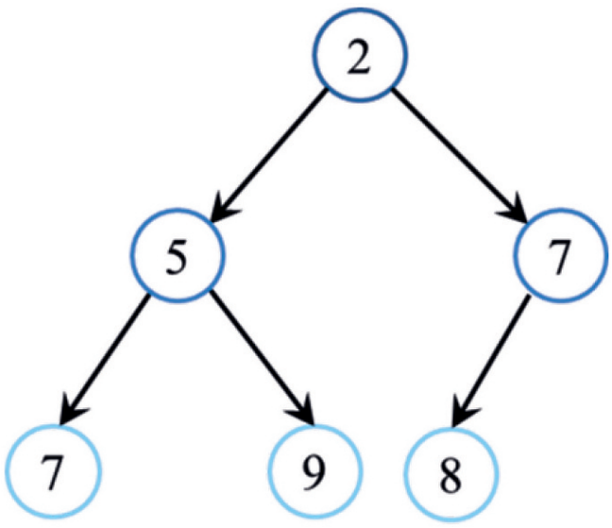


Fig. 7 - Esempio di struttura dati heap con sei elementi.
Fig. 7 - Example of heap data structure with six items.

03_IF_10_2016 pag_769_788__ 02/11/16 09:06 Pagina 776

Un heap binario è un albero binario quasi completo, ovvero un grafo non orientato che presenta le seguenti caratteristiche:

- due vertici sono connessi un solo cammino;
- il nodo iniziale è detto radice;
- ogni nodo esclusa la radice ha un solo padre;
- un nodo senza figli è detto foglia;
- i nodi possono avere al più due figli;
- completo fino al penultimo livello;
- ad ogni nodo viene attribuita una chiave (d[v]);
- il padre ha una chiave minore o uguale a quella dei figli.

Individuare quindi nell'insieme Q il nodo a cui corrisponde il valore d[v] minimo risulta molto immediato, in quanto corrisponde alla radice dell'albero.

Sono state impostate tre tipologie di operazioni sull'heap binario.

Inserimento di un nodo: nel momento in cui si vuole inserire un nuovo nodo v con valore d[v] all'interno dell'heap, questo viene inserito come foglia più a destra nell'ultimo livello. Se d[v] è minore di quello associato al padre, si applica una procedura di "riaggiustamento" che consiste nello scambiare il nodo appena inserito con il padre (fig. 8).

Ricerca del minimo: l'etichetta di valore minimo è sempre quella associata alla radice, quindi per individuare il nodo con d[v] minimo basta considerare il nodo che si trova alla radice (fig. 9).

Estrazione del minimo: si deve sostituire alla radice il valore della foglia più a destra nell'ultimo livello dell'heap, dopodiché la foglia viene eliminata riducendo di uno il numero di nodi dell'albero e se l'etichetta associata al nuovo nodo è maggiore di quella di uno dei figli, si procede con la procedura di riaggiustamento (fig. 10).

4. Sviluppo del software ed analisi dei risultati

Modellizzato il problema e scelto l'algoritmo per la ricerca del cammino minimo si è deciso di implementare il progetto su Microsoft Access utilizzando come linguaggio di programmazione Visual Basic[2].

- *a node with no children is called leaf;*
- *nodes can have at the most two children;*
- *complete until the penultimate level;*
- *every node is given a key (d[v]);*
- *the parent has a key less than or equal to that of its children.*

Therefore, locating the node that matches the minimum d[v] value in the Q set is very straightforward, since it corresponds to the root of the tree.

Three types of operations have been set up on the binary heap.

Inserting a node: *when you want to insert a new v node with value d[v] within the heap, it is inserted as the rightmost leaf in the last level. If d[v] is less than that associated with the father, a "readjustment" procedure is applied which consists in exchanging the newly inserted node with the parent (fig. 8).*

Searching of minimum: *the minimum value label is always the one associated with the root, hence in order to find the node with minimum d[v] just consider the node at the root (fig. 9).*

Extraction of minimum: *the value of the rightmost leaf in the last level of the heap needs to be replaced, then the leaf is eliminated by reducing by one the number of nodes in the tree, and if the label associated with the new node is greater than that of one of the children, one proceeds with the readjustment process (fig. 10).*

4. Software development and analysis of the results

Having modelled the problem and chosen the algorithm for finding the shortest path it was decided to implement the project on Microsoft Access using Visual Basic as the programming language[2].

A simple graphical interface of the Blue Train programme allows the user to enter all the data concerning the journey, namely:

- *origin stations;*
- *destination station;*
- *day and departure time slot;*
- *need for equipped place.*
- *method of calculation.*

---

[2] Il software individua soluzioni con una velocità di elaborazione del codice pari a 0,68 secondi per relazione, se utilizzato con un computer avente le seguenti caratteristiche:
- Processore: Intel(R) Core(TM) i7 – 3770 CPU @ 3.40 GHz 3.40 GHz;
- Classificazione: 5,9 indice prestazioni Windows;
- Memoria istallata (RAM): 8,00 GB;
- Tipo di sistema: sistema operativo a 64 bit.

[2] The software identifies solutions with a code processing speed of 0.68 seconds per connection, if used with a computer having the following features:
- *Processor: Intel (R) Core (TM) i7 - 3770 CPU @ 3.40 GHz 3.40 Ghz;*
- *Classification: Windows performance index 5.9;*
- *Installed memory (RAM): 8.00 GB;*
- *System type: 64-bit operating system.*

03_IF_10_2016 pag_769_788__ 02/11/16 09:06 Pagina 777

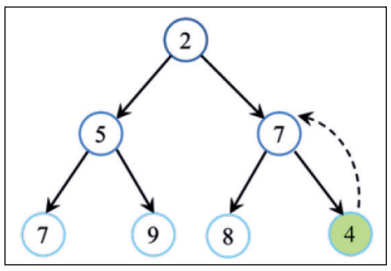


Fig. 8 - Esempio inserimento nodo.
Fig. 8 - Example of node insertion.

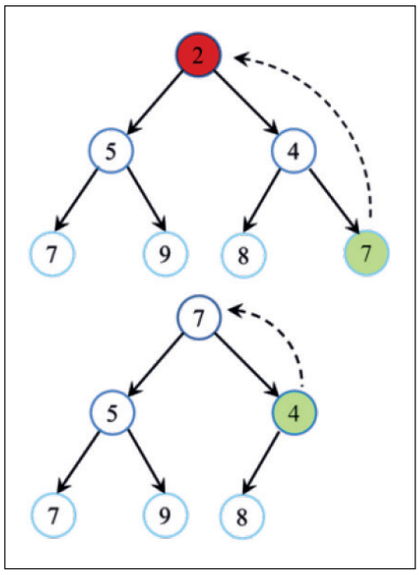


Fig. 9 - Esempio ricerca minimo.
Fig. 9 - Example of search for minimum.

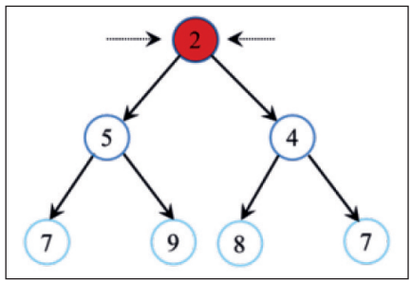


Fig. 10 - Esempio eliminazione minimo.
Fig. 10 - Example of elimination of minimum.

Una semplice interfaccia grafica del programma Treno Blu consente all'utente di inserire tutti i dati riguardanti il viaggio e cioè:

– stazioni di origine;
– stazione di destinazione;
– giorno e fascia oraria di partenza;
– necessità di posto attrezzato.
– criterio di calcolo.

Nella parte inferiore della maschera, una volta terminate le operazioni di calcolo, vengono visualizzati i percorsi trovati con indicazione per ogni singola tratta del numero del treno, dell'impresa ferroviaria, della categoria, della stazione di partenza, dell'ora di partenza, della stazione di arrivo, dell'ora di arrivo e della presenza sul treno di posti attrezzati. Inoltre vengono visualizzati per ogni viaggio il numero di cambi e la durata complessiva (fig. 11).

A titolo di esempio, per dimostrare l'efficacia del software si sono messe a confronto alcune soluzioni offerte da Treno Blu con quelle standard del software di Trenitalia.

Per garantire solo i percorsi con cambio in un tempo maggiore a 15 minuti si sono cercate le soluzioni di viaggio tra Roma Termini e Rovigo dalle ore 8. Come

At the bottom of the window, once calculations are completed, the paths found are displayed indicating the train number for each individual route, the railway undertaking, the category, the departure station, departure time, arrival station, time of arrival and presence of equipped places on the train. The number of changes and the total duration are also displayed for each journey (fig. 11).

By way of example, to demonstrate the effectiveness of the software some solutions offered by Blue Train are compared with standard ones of the Trenitalia software.

To ensure only routes with change exceeding a period of 15 minutes, travel options between Roma Termini and Rovigo from 8 a.m. were searched. As shown in fig. 12,

itir	Treno	IF	Categoria	Stazione Partenza	Ora Par	Stazione Arrivo	Ora Arr	Att	cambi	durata
1	09410	TRENITALIA SpA -	EUROSTAR ITALIA	ROMA TERMINI	08:50:00	ROVIGO	11:44:00	X	0	2:54
3	09518	TRENITALIA SpA -	EUROSTAR ITALIA	ROMA TERMINI	09:20:00	BOLOGNA C. LE	11:35:00	X	1	3:48
3	02232	TRENITALIA SpA -	REGIONALE	BOLOGNA C.LE	12:20:00	ROVIGO	13:08:00		1	3:48
10	00588	TRENITALIA SpA -	INTERCITY	ROMA TERMINI	10:30:00	ROVIGO	15:32:00		0	5:02
5	09416	TRENITALIA SpA -	EUROSTAR ITALIA	ROMA TERMINI	10:35:00	BOLOGNA C.LE	12:52:00	X	1	3:33
5	02234	TRENITALIA SpA -	REGIONALE	BOLOGNA C.LE	13:20:00	ROVIGO	14:08:00		1	3:33
6	09524	TRENITALIA SpA -	EUROSTAR ITALIA	ROMA TERMINI	11:20:00	BOLOGNA C. LE	13:35:00	X	1	3:38

Fig. 11 - Software Treno Blu: percorsi Roma Termini-Rovigo dalle ore 8.
Fig. 11 - Blue Train Software: Roma Termini-Rovigo routes from 8 a.m.

03_IF_10_2016 pag_769_788__ 02/11/16 09:06 Pagina 778

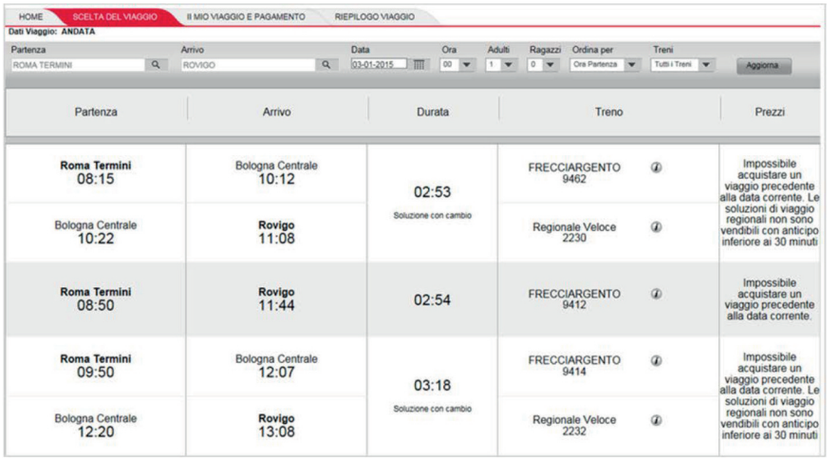


Fig. 12 - Software "Treno Blu": percorsi Roma Termini-Rovigo dalle ore 8.
Fig. 12 - "Blue Train" Software: Roma Termini-Rovigo routes from 8 a.m.

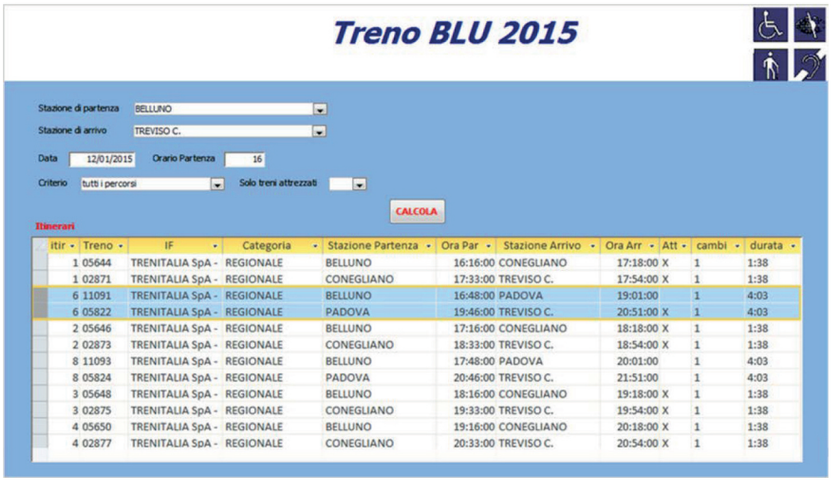


Fig.13 - Software Treno Blu: percorsi Belluno-Treviso dalle ore 16.
Fig. 13 - Blue Train Software: Belluno-Treviso routes from 4 p.m.

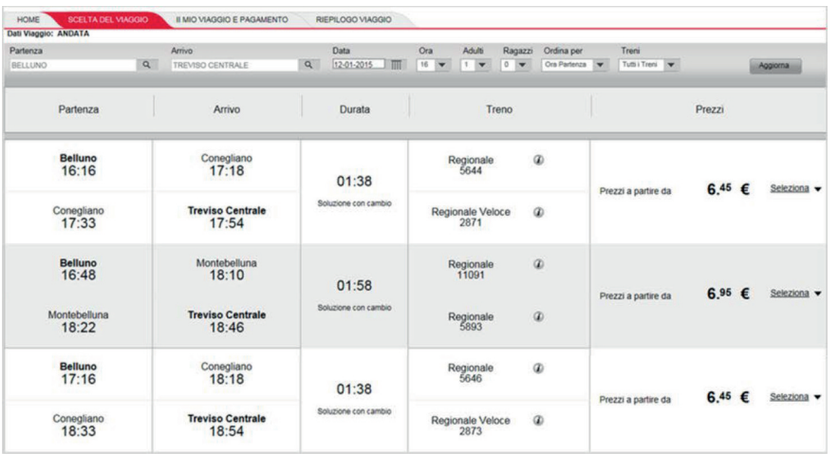


Fig. 14 - Software Trenitalia: percorsi Belluno-Treviso dalle ore 16.
Fig. 14 - Trenitalia Software: Belluno-Treviso Routes from 4 p.m.

Blue Train's solutions do not include those departing from Roma Termini at 8:15 and at 9:50, identified by Trenitalia software, as they foresee a change in 10 and 13 minutes respectively.

The inclusion of locations only of stations offering assistance is demonstrated by the Belluno - Treviso solution of 16:48 with change of train in Padua, while the Trenitalia software offers the 16:48 solution with a change of train in Montebelluna (station lacking assistance) (figures 13 and 14).

To verify whether the Blue Train suggests "excellent" solutions considering trains of all railway undertakings, trips from Roma Tiburtina to Brescia at 8 a.m. were searched; it can be observed how the first journey choice given by the programme provides departure from Roma Tiburtina at 8:55 with train 9914 of Nuovo Trasporto Viaggiatori, change at Bologna and arrival in Brescia, scheduled at 12:50 (fig. 15). Such a solution that is obviously not found on the Trenitalia website proposing as first solution a trip with change in Milan and arrival in Brescia at 13:21 (fig. 16).

Verification of the search criteria of only equipped trains is shown for the Orte-Roma Termini at 8 a.m.: the first train suggested is that of 8:16, while that of 8:01 indicated by the Trenitalia system is excluded from the solutions as it is not equipped (figures 17 and 18).

5. Conclusions

The computer system designed to search travel options intended for people with reduced mobility is based on the following constraints:

- *it includes trains of all railway undertakings;*
- *it includes only stations that provide assistance;*
- *it foresees a train change with an interval of over 15 minutes;*
- *if requested it finds only passenger trains equipped for wheelchair users.*

Rete Ferroviaria Italiana is considering testing of such programme at the

03_IF_10_2016 pag_769_788__ 02/11/16 09:06 Pagina 779

riportato in fig. 12, le soluzioni offerte da Treno Blu non includono quelle con partenza da Roma Termini alle 8:15 e alle 9:50, individuate dal software di Trenitalia, in quanto prevedono un cambio rispettivamente in 10 e 13 minuti.

L'inclusione nei percorsi solamente di stazioni che offrono assistenza, è dimostrata dalla soluzione Belluno – Treviso delle 16:48 con cambio treno a Padova, mentre il software di Trenitalia offre la soluzione delle 16:48 con cambio treno a Montebelluna (stazione priva del servizio di assistenza) (figg. 13 e 14).

Per verificare che Treno Blu suggerisca le soluzioni "ottime" prendendo in considerazione i treni di tutte le Imprese Ferroviarie sono stati cercati i viaggio da Roma Tiburtina a Brescia a partire dalle ore 8; si può osservare come la prima soluzione di viaggio restituita dal programma preveda la partenza da Roma Tiburtina alle 8:55 con il treno 9914 di Nuovo Trasporto Viaggiatori, cambio a Bologna ed arrivo a Brescia previsto per le 12:50 (fig. 15). Tale soluzione che non viene ovviamente individuata sul sito di Trenitalia che propone come prima soluzione un viaggio con cambio a Milano e un arrivo a Brescia alle 13:21 (fig. 16).

La verifica del criterio di ricerca dei soli treni attrezzati è mostrato per la tratta Orte-Roma Termini delle ore 8: il primo treno suggerito è quello delle 8:16, mentre viene escluso dalle soluzioni quello delle 8:01 indicato dal sistema di Trenitalia non essendo attrezzato (figg. 17 e 18).

5. Conclusioni

Il sistema informatico realizzato per ricercare soluzioni di viaggio destinate a persone a ridotta mobilità si basa sui seguenti vincoli:

– include treni di tutte le imprese ferroviarie;

Fig. 15 - Software Treno Blu: percorsi Roma Tiburtina-Brescia dalle ore 8.
Fig. 15 - Blue Train Software: Roma Tiburtina-Brescia routes from 8 a.m.

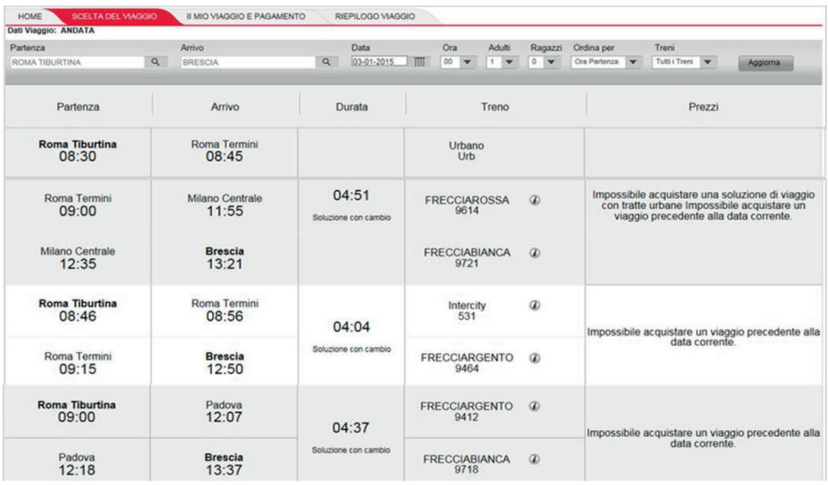


Fig. 16 - Software Trenitalia: percorsi Roma Tiburtina-Brescia dalle ore 8.
Fig. 16 - Trenitalia Software: Roma Tiburtina-Brescia routes from 8 a.m.

Fig. 17 - Software "Treno Blu": percorsi Orte-Roma Termini dalle ore 8 con treno attrezzato.
Fig. 17 - "Blue Train" software: Orte-Roma Termini routes from 8 a.m. with equipped train.

03_IF_10_2016 pag_769_788__ 02/11/16 09:06 Pagina 780

– include solo stazioni che offrono servizio di assistenza;
– prevede un cambio treno con più di 15 minuti di intervallo;
– se richiesto individua solo treni attrezzati al trasporto passeggeri su sedia a rotelle.

Rete Ferroviaria Italiana sta valutando di sperimentare tale programma presso le Sale Blu[3] a sussidio degli operatori nella prenotazione dei servizi di assistenza per poi eventualmente trasformarlo in un software disponibile online. Inoltre tale progetto potrà costituire la base per lo sviluppo di un software di simulazione finalizzato a valutare l'effetto delle modifiche di orario sulla mobilità ferroviaria delle persone con disabilità e definire azioni preventive per l'adeguamento del circuito di assistenza con l'eventuale inserimento di nuove stazioni.

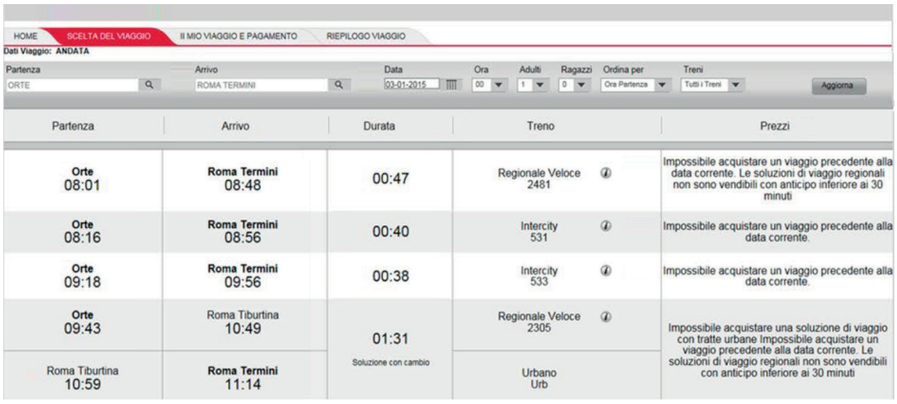


Fig. 18 - Software Trenitalia: percorsi Orte-Roma Termini dalle ore 8.
Fig. 18 - Trenitalia software: Orte-Roma Termini routes from 8 a.m.

Possibili sviluppi futuri del modello potranno riguardare invece:

– l'interfacciamento del motore orario con i sistemi di gestione della circolazione in modo da calcolare le alternative di viaggio sulla base delle condizioni reali di circolazione dei treni e non sul programmato teorico: si dovranno quindi introdurre nel modello gli eventuali ritardi dei treni che porteranno ad un allungamento degli archi del grafo in tempo reale,
– l'inclusione nel calcolo del percorso dei tratti interni alle stazioni di origine/destino; tali archi dovranno collegare l'ingresso del Fabbricato Viaggiatori ai marciapiedi su cui si attestano i treni suggeriti e consentiranno di includere nell'output finale il grado di accessibilità delle stazioni di origine/destino evidenziando percorsi privi di ostacoli che le persone a ridotta mobilità possono percorrere in autonomia.

Blue Lounges[3] in aid of the operators in the reservation of assistance and eventually of turning it into a software available online.

Furthermore, this project will form the basis for developing simulation software aimed at evaluating the effect of the changes of train timetable on the rail mobility of persons with disabilities and define preventive actions for the adjustment of the assistance circuit with the possible inclusion of new stations.

Possible future developments of the model may instead concern:

– *motor time interfacing with traffic management systems to calculate travel alternatives on the basis of actual train circulation conditions and not on the theoretical planned; train delays that will lead to lengthening of the arcs of the graph in real time will then have to be introduced in the model.*
– *inclusion in the route calculation of internal origin/destination stations stretches; these arcs will have to connect the Passenger Building entrance to the sidewalks on which the trains suggested stop and will include the degree of accessibility of origin/destination stations in the final output highlighting paths free of obstacles along which people with reduced mobility can travel independently.*

---

[3] Le Sale Blu rappresentano il punto di riferimento per l'organizzazione del servizio offerto nelle stazioni del circuito di assistenza PRM svolto RFI. Sono presenti in 14 principali stazioni e assicurano:
– informazioni e materiale illustrativo sul servizio di assistenza;
– la prenotazione del servizio;
– la guida in stazione e l'accompagnamento al treno;
– l'accoglienza in treno all'arrivo in stazione e la guida fino all'uscita di stazione o ad altro treno corrispondente;
– la messa a disposizione, su richiesta, della sedia a ruote;
– la salita e la discesa con carrelli elevatori per i viaggiatori su sedia a ruote;
– l'eventuale servizio gratuito, su richiesta, di portabagagli a mano.

[3] The Blue Lounges (Sale Blue) represent a reference point for the organisation of the service offered in the stations of the PRM assistance circuit carried out by RFI. They can be found in 14 main stations and ensure:
– *information and illustrative material on assistance;*
– *booking of the service;*
– *guide in the station and accompaniment to the train;*
– *welcome in the train upon arrival at the train station and guide to the exit at the station or to other corresponding train;*
– *provision of wheelchair upon request;*
– *boarding and getting off by forklift trucks for travellers in wheelchairs;*
– *any free luggage hand carrying service, on request.*

03_IF_10_2016 pag_769_788__ 02/11/16 09:06 Pagina 781

BIBLIOGRAFIA - *REFERENCES*

[1] Regolamento CE 1371/2007, *"Diritti e obblighi dei passeggeri nel trasporto ferroviario"*.

[2] K. Thulasiraman, M.N.S. Swamy, *"Graphs: Theory and Algorithms"*, J. Wiley, 1992.

[3] M.T. Goodrich, R. Tamassia, *"Strutture dati e algoritmi in Java"*, Zanichelli, 2007.

[4] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *"Introduction to Algorithms"*, Seconda Edizione, MIT Press and McGraw-Hill, 2001.

[5] A.A. Bertossi, A. Montresor, *"Algoritmi e strutture di dati (III edizione)"*, CittàStudi, 2014.

Web – grafia

[A] Quadri Orario RFI - https://prm.rfi.it/qo_prm/

[B] Soluzioni di viaggio http://www.trenitalia.com/; http://www.italotreno.it/

[C] Approccio tempo espanso/dipendente - http://www.iti.uni-karlsruhe.de/extra/publications/mswz-tima-06.pdf

[D] Algoritmo di Dijkstra - http://ioi.di.unimi.it/dijkstra.pdf

[E] Metodi di implementazione della coda di priorità - http://twiki.di.uniroma1.it/pub/ASD/AlgoritmiEStruttureDati2010_2011.pdf