



Generazione automatica di orari ed applicazioni di algoritmi di programmazione per lo studio di capacità e perturbazioni di linee ferroviarie

*Scheduling algorithms for rail operations and
the automatic generation of timetables: application
for railway capacity and perturbation evaluation*

Dott. Ing. Nicola COVIELLO^(*)
Prof. Ing. Bruno DALLA CHIARA^(**)
Prof. Ing. Stefano RICCI^(***)

Sommario - L'articolo presenta uno strumento per l'analisi di capacità di reti ferroviarie. Questo strumento si compone di un primo livello applicativo, che può essere considerato di tipo microscopico e permette di simulare numericamente la marcia dei treni calcolando l'occupazione delle sezioni di blocco. Un secondo livello, di tipo mesoscopico, fa uso di dati di input aggregati - per esempio tempi di percorrenza tra stazioni consecutive o minimi distanziamenti ammessi dal sistema - che possono essere automaticamente calcolati dal microsimulatore come anche inseriti dall'utente. Tramite tali dati aggregati, un algoritmo di programmazione produce degli orari fattibili, ottimizzati secondo dati parametri di qualità. Dal momento che il processo è automatizzato ed abbastanza veloce come tempi computazionali, può essere adoperato per produrre insiemi di orari fattibili e per eseguire analisi di capacità basate sugli orari. È altresì presentata l'implementazione dell'algoritmo in uno strumento dedicato alle analisi di perturbazione, strumento basato sulla simulazione ad eventi discreti. Esso applica determinate perturbazioni (ritardi, incidenti, anomalie) ad un orario di partenza, il quale viene riprogrammato per risolvere eventuali conflitti di circolazione. Tale simulazione valuta quindi la robustezza del sistema. Può essere inoltre valutata l'efficacia degli algoritmi di riprogrammazione, i quali rappresentano diverse strategie che possono essere adottate da parte delle entità preposte al controllo della circolazione.

Summary - The paper presents a tool for the assessment of the capacity of railway networks. This tool includes a first definition level that can be considered as microscopic, which permits train-runs to be simulated in a numerical way, through the identification of the occupation of block sections; a second mesoscopic level then uses aggregated data (e.g., the run times between stations or the minimum admitted headways), which can be calculated automatically by the micro-simulator or entered directly by the user, as input.

A scheduling algorithm, using the aggregated data, produces feasible timetables, optimised according to given quality parameters.

Since the procedure is automated and the computation phase is rather quick, it can be used to generate sets of feasible timetables and to perform timetable-based capacity assessments. Its implementation, within a perturbation analysis, is based on a discrete-event simulation core.

This core applies any perturbation (delay, accident, anomaly) to a given timetable, which is re-arranged in order to solve any traffic conflict and to assess the robustness of the timetable design.

The effectiveness of the rescheduling algorithms, which can be used to simulate different strategies that could be adopted by dispatchers, is also evaluated.

^(*) Studente di Dottorato, Politecnico di Torino, Ingegneria, Dip. DIATI – Sistemi di Trasporto e Università di Roma “La Sapienza”, DICEA – Area Trasporti.

^{(**) Professore Associato, Politecnico di Torino, Ingegneria, Dip. DIATI – Sistemi di Trasporto.}

^{(***) Professore Associato, Università di Roma “La Sapienza”, DICEA – Area Trasporti.}

^{(*) Ph.D. student, Politecnico di Torino, Engineering, Dept. DIATI – Transport Systems and Università di Roma “La Sapienza”, DICEA – Transport Area.}

^{(**) Associate Professor, Politecnico di Torino, Engineering, Dept. DIATI – Transport Systems.}

^{(***) Associate Professor, Università di Roma “La Sapienza”, DICEA – Transport Area.}

1. Introduzione

1.1. Descrizione dell'articolo

Questo articolo descrive l'implementazione di un algoritmo di programmazione (*scheduling*) della circolazione ferroviaria e la sua applicazione nel campo delle valutazioni di capacità. Dal momento che tale studio è tuttora in corso, è opportuno puntualizzare che i risultati qui discussi rappresentano i primi passi consolidati di questa ricerca, prevedendo già ulteriori sviluppi.

L'algoritmo di programmazione è implementato in un ambiente di simulazione mesoscopico dell'esercizio ferroviario, facente parte di un originale programma di calcolo sviluppato per essere uno strumento efficace ma al tempo stesso di uso abbastanza semplice per effettuare analisi di capacità. Questa applicazione è stata sviluppata dal primo autore, come parte della sua attività di ricerca di Dottorato.

Anche se questo articolo non riguarda il programma di calcolo di per sé, verrà fornita una sua descrizione generale, dal momento che essa inquadra adeguatamente l'obiettivo per cui il modello è stato sviluppato. Infatti tale ambiente di simulazione è volto ad effettuare valutazioni, basate su orari fattibili (analisi *timetable-dependent*), dell'influenza che diversi parametri, sia infrastrutturali sia d'esercizio, possono avere sulla capacità.

Come riportato dalle fonti bibliografiche in materia, le tecniche per analizzare e calcolare la capacità delle linee possono essere divise in tre grandi categorie:

- metodi sintetici ed analitici, che modellizzano il sistema ferroviario in maniera semplice, tramite espressioni matematiche;
- metodi di ottimizzazione, basati cioè sulla ricerca di orari ottimamente saturati;
- simulazioni, che mettono a disposizione modelli per rappresentare la realtà e per validare i parametri di orario.

Mentre il calcolo analitico è usualmente preferito per ottenere una prima indicazione sulla capacità attraverso formule matematiche o espressioni algebriche, i metodi ottimizzativi sono basati su modelli d'esercizio, che tentano di calcolare e minimizzare i ritardi per determinati orari fattibili. Diversamente, gli ambienti di simulazione sono strumenti che, di norma, generano gli orari attraverso equazioni che sono definite nel tempo lungo intervalli finiti [1]. Il processo di simulazione sviluppa una grande varietà di dati di *output*, i quali includono vari tipi di diagrammi e, cosa di primario interesse, orari grafici. Essi permettono di saturare la capacità della linea sulla base del modello di servizio definito dall'utente, inserendo manualmente ulteriori tracce in un orario esistente.

È opportuno evidenziare come l'obiettivo del metodo presentato in questo articolo sia piuttosto diverso da quelli di molti pacchetti di simulazione ferroviaria com-

1. Introduction

1.1. Paper description

This paper presents the implementation of a rescheduling algorithm for railway traffic and its application in the capacity assessment field. It is necessary to point out that the results discussed hereafter represent the first consolidated steps of the research; more developments can be envisaged since this study is still underway.

The scheduling algorithm is implemented within a mesoscopic simulation environment for railway operation and both have been developed within an original calculation package with the target of providing an effective – yet, at the same time, rather simple tool – that can be used to carry out capacity analyses. This application was conceived and developed by the first author as a part of his Ph.D. research activity.

Even though this paper does not include a description of the calculation package itself, a general explanation has been provided, since it is able to properly frame the main target for which the model was conceived and implemented. This simulation environment is in fact devoted to evaluating, in a timetable-dependent way, the influence that different parameters, whether infrastructural or operational, can have on capacity.

As reported in the relevant literature, the techniques used to analyse and calculate the carrying capacity of trains can be divided into three large categories, namely:

- synthetic and analytical methods, which simply model the railway infrastructure, plus some operational and maintenance features, according to the used approach, through mathematical expressions;*
- optimisation methods, i.e. those based on the search for the optimal saturated timetables;*
- simulations, which provide models and step-by-step emulating instruments to represent the reality, as closely as possible, and to validate the timetable data.*

Although analytical computing is usually performed to obtain a first rough indication on the traffic capacity through either mathematical formulas or algebraic expressions, optimisation methods are based on operational models, with which their authors attempt to calculate and minimise delays within feasible timetables. In the latter case, the simulation environments are tools which usually generate the graphic timetables through equations that are specifically defined in time at finite intervals [1]. The simulation processes generate a large variety of output data, including diagrams of various types and, mainly, graphic timetables. They also allow the carrying capacity of a line to be saturated on the basis of the service model defined by the user; this is achieved by manually inserting additional train paths into an already existing timetable.

It is worth mentioning that the target of the method presented in this paper is rather different from those of the

merciali. Quest'ultimi tipicamente mirano a modellizzare l'intero ambiente ferroviario al fine di ottenere una fedele rappresentazione della realtà, utile per valutare scenari reali o pianificati in modo accurato. D'altra parte, essi richiedono un gran quantitativo di dati di *input* nonché parecchio tempo da dedicare alla costruzione dei modelli: tali sforzi possono spesso essere sproporzionati rispetto ai risultati perseguiti. È questo il caso, per esempio, di analisi preliminari, tese a fornire solo intervalli indicativi, che quindi non necessitano di un accurato e meticoloso modello infrastrutturale. Inoltre, tali programmi di simulazione spesso non forniscono strumenti esplicitamente dedicati ad analisi di capacità, rendendo dunque necessarie laboriose operazioni di *post-processing*, spesso complicate dal fatto che i dati di processo sono memorizzati in formati proprietari e non sono quindi facilmente accessibili per gli utenti.

1.2. Applicazioni per analisi di capacità

Modellizzare l'esercizio ferroviario tramite un approccio mesoscopico permette di contenere significativamente lo sforzo computazionale richiesto, semplificando nello stesso tempo l'inserimento dei dati di input da parte dell'utente. Un modello mesoscopico fa uso di insiemi di dati di ingresso già aggregati; se da un lato ciò introduce approssimazioni non trascurabili - le quali sono in ogni caso quantificabili - dall'altro permette di implementare efficacemente applicazioni sofisticate come quelle fornite da un algoritmo di programmazione automatico.

Questa funzione permette di configurare uno strumento in grado di effettuare analisi di capacità basate sugli orari. È opportuno sottolineare che tale compito può essere svolto con qualsiasi microsimulatore ferroviario che permetta, tramite apposite funzioni, di progettare e modificare orari. Ad ogni modo, per le ragioni discusse nel seguito, questo approccio alla valutazione della capacità richiede di generare ed analizzare una gran quantità di orari - decine se non centinaia - i quali non possono evidentemente essere costruiti manualmente. Questo è d'altra parte il compito principale dell'algoritmo di programmazione automatico, in grado di generare un orario fattibile - partendo da certi dati di input e rispettando opportuni vincoli - in poche decine di secondi.

Un approccio alla valutazione di capacità basato sugli orari di fatto permette di considerare tutti i parametri significativi che caratterizzano l'esercizio ferroviario, dal momento che i loro effetti sono direttamente condensati dall'orario in sé. Diversamente, tutti i metodi analitici e indipendenti dall'orario si basano su precise assunzioni, le quali trascurano, a seconda dei casi, taluni parametri fornendo dunque una visione parziale del fenomeno.

Inoltre i metodi analitici spesso sovrastimano gli effetti prodotti da una variazione di un certo parametro. Per esempio, potrebbe essere abbastanza semplice stimare i benefici, in termini di incremento di capacità, dovuti alla riduzione della lunghezza delle sezioni di blocco in

available commercial railway simulation packages, some tens of which are available throughout the world, of which only a few are well-known and widely diffused. These packages generally have the aim of modelling the entire railway environment, even when it is limited to a few kilometres of track, in order to obtain a realistic representation of the reality, which is useful for an accurate evaluation of the actual or planned scenarios. However, they inevitably require a huge amount of input data and a long time to construct the models: such efforts may often result disproportionate with respect to the expected output, even more so when dealing with single lines rather than an entire network, for which this time is amortised over the years by, for example, a railway manager. This is the case, for example, of preliminary analyses, which should provide only indicative ranges of values, while accurate and meticulous models of the line are not required. Furthermore, such simulation packages do not usually provide tools devoted explicitly to the capacity assessment, and as a result laborious post-processing operations, which are often complex due to the proprietary formats of the stored, non-user friendly processed data, are required.

1.2. Applications for capacity analysis

Modelling railway operations through a mesoscopic approach makes it possible to significantly contain the relevant computation effort, and, at the same time, to simplify the input data asked for by the user. A mesoscopic model makes use of a set of already aggregated input data; on one hand, although it introduces non-negligible approximations - which are computable - on the other hand, it permits smart applications to be effectively implemented, like those supported by the automatic scheduling algorithm.

This function allows a tool to be set up that is able to perform any timetable-based capacity analysis. It is worth mentioning that such a task could be performed with almost any railway microsimulation tool that is able to easily handle timetable-editing functions. However, for the reasons presented hereafter, this approach to capacity evaluation requires a large number of timetables (tens or even hundreds) to be produced and analysed, which are evidently not suitable for editing by hand. This, on the other hand, is the main task of the automatic scheduling algorithm, which allows a feasible timetable to be arranged - starting from certain input data and respecting given constraints - in just a few tens of seconds.

A timetable-based capacity evaluation approach actually permits all the significant parameters that characterise rail operations to be taken into account, since their effects are directly condensed by the timetable itself. All the available analytical and timetable-independent methodologies are instead based on precise assumptions that, case by case, neglect some parameters thus giving only a partial view of the phenomenon.

una certa tratta, nel caso che producano un calo del minimo distanziamento ammesso. D'altra parte tale calcolo stimerebbe di fatto il beneficio reale solo se quella linea fosse pienamente saturata, cioè se i treni si susseguissero l'un l'altro con il minimo distanziamento permesso dal sistema di segnalamento. Questa è certamente una situazione limite, indicativa di una condizione di circolazione congestionata e probabilmente corrispondente solo a poche finestre temporali all'interno delle 24 ore. Pertanto i reali benefici dell'accorciamento delle sezioni di blocco possono essere stimati solo attraverso un orario, in grado di determinare l'efficacia di un certo provvedimento, infrastrutturale od operativo.

1.3. Rassegna bibliografica

La valutazione di capacità tramite orari è corrente oggetto di ricerca. Nell'ultima decade infatti i programmi di simulazione commerciale hanno rivelato la potenzialità di tale approccio, sollevando al tempo stesso la necessità di strumenti esplicitamente dedicati a tal fine. Per questa ragione, anche se è ancora materia di ricerca corrente, numerosi spunti sono presenti nella letteratura disponibile.

È opportuno sottolineare che l'idea di effettuare analisi di capacità usando gli orari è stata suggerita da alcuni raffinati metodi analitici. Quest'ultimi, generalmente sviluppati per analizzare l'esercizio su ferrovie a singolo binario, introducono vincoli legati all'esercizio nelle formule complessive. Tali vincoli sono rappresentati da parametri appropriati, i quali modellizzano l'influenza dell'orario stesso, cioè dei tempi morti dovuti agli incroci. Di conseguenza, la definizione di tali parametri non è univoca, ed essi possono essere forniti come intervalli di valori fattibili (o possibili). Anche i risultati dunque prendono la forma di intervalli di capacità. Questo è il caso, per esempio, dello studio presentato in [2], relativo all'analisi di due diversi tipi di linee a singolo binario. In particolare, questo lavoro [2] ha integrato l'approccio basato su intervalli di parametri operativi con la metodologia già presentata in [3].

Questi metodi di fatto introducono alcune variazioni probabilistiche all'interno di formule strettamente probabilistiche. Diversamente, in [4] una metodologia valutativa è implementata nel pacchetto di calcolo SAMFOST tramite un approccio probabilistico. In questo modo, il modello mette a fuoco svariati interventi utili per migliorare la qualità del servizio, interventi qualificati da un valore statistico della loro attuale efficacia. In ogni caso, anche se tale approccio presenta interessanti risultati quantitativi e qualitativi che inquadrono un'originale metodologia d'analisi, presenta ancora alcune significative assunzioni di base.

Questi approccio introducono il problema di valutare questi intervalli di parametri o le loro distribuzioni statistiche. Tale compito può essere effettuato analizzando un certo numero di orari, che possono essere sia reali - effettivamente impiegati nell'esercizio giornaliero - sia virtuali

Moreover, such methods often overestimate the effects that a variation of certain parameters could produce. For example, it could be rather simple to estimate the benefits, in terms of capacity improvement, resulting from the reduction of the length of the block sections on a certain stretch, which would result in a drop in the minimum admitted headway. However, this calculation would actually only estimate the real benefit if that line were fully saturated, i.e. when trains follow each other with the minimum headway allowed by the installed signalling system. This represents a limit situation, which is indicative of congested traffic conditions, but which usually corresponds to only a few time slots per day. Therefore, the real benefits of a block section reduction can only be estimated through a timetable that is able to determine the effectiveness of a given measure (infrastructural or operational, including the actual performances of the rolling stock).

1.3. Literature review

Capacity assessment, based on a timetable-dependent approach, is a research field of topical interest. In fact, over the last decade, commercial simulation packages have emerged that reveal the potentiality of this approach, but the need for explicitly devoted tools has also arisen. For this reason, even though it is a matter of current research, several ideas are already present in the available literature.

It is worth pointing out that the idea of carrying out a capacity assessment using timetables arose from the use of some refined analytical evaluation methodologies. These methodologies, which were generally developed to analyse operations on single-track railways, introduce operational constraints into the overall formulas. These constraints are represented by the necessity of introducing appropriate parameters that, in some way, are able to model the influence of timetables, e.g. the times lost due to train crossings. Consequently, the definition of such parameters is not univocal, and they should therefore be introduced as a range of feasible (or possible) values. The results will therefore also take the shape of capacity ranges. This is the case, for instance, of the study presented in [2], which refers to the analysis of two different types of single-track lines. In [2], the authors in particular based their approach on the range of operational parameters presented in [3].

These methodologies actually introduce some probabilistic variations into strictly deterministic formulas. Instead, in [4], an evaluation methodology has been implemented in the SAMFOST tool through a probabilistic approach. This model points out a number of interventions that are useful for improving the quality of the service, and which are characterised by a statistical value of their actual effectiveness. However, even though this concept presents various interesting qualitative and quantitative results, which effectively outline an original methodology for the

e fattibili - costruiti quindi per analizzare possibili situazioni future - suggerendo così la prima bozza del metodo presentato in questo articolo.

Tale compito coinvolge questioni di ottimizzazione al fine di ottenere orari che siano non solo fattibili (privi di conflitti di circolazione), ma che rispettino altresì dati parametri di qualità. Questo problema richiede strumenti matematici abbastanza sofisticati da utilizzarsi in maniera appropriata, ed è stato studiato da vari ricercatori con ottiche diverse. Un compendio del relativo stato dell'arte può essere trovato in [5]: in particolare nei capitoli 8 e 9 sono proposti e discussi numerosi modelli di ottimizzazione per gli orari ferroviari.

In questa ricerca un primo tentativo per risolvere il problema della generazione automatica degli orari ha visto l'applicazione di un algoritmo genetico (GA), seguendo le indicazioni di numerose fonti bibliografiche. Per esempio in [6] è proposto un metodo per la generazione di orari basato sull'algoritmo genetico, fornendo una panoramica globale delle istanze matematiche che caratterizzano il problema. Diversi approcci alla sua soluzione sono inoltre discussi e confrontati, dimostrando l'efficacia del modello proposto. Analogamente in [7] gli algoritmi genetici sono adoperati per risolvere un *railway timetabling problem* legato all'allocazione dei percorsi dei treni all'interno di una rete complessa (non riguardando dunque la programmazione delle tracce dei treni lungo una singola linea), evidenziando come sia necessario introdurre dei criteri euristici per migliorare le prestazioni dell'algoritmo. Anche in [8] vengono raggiunte analoghe conclusioni.

Dopo aver presentato e discusso i programmi di simulazione ferroviaria disponibili sul mercato, in [9] è presentato un algoritmo euristico integrato in uno strumento di calcolo (pacchetto MOM) in grado di calcolare orari cadenzati ottimi. L'algoritmo è successivamente generalizzato anche ad orari non cadenzati. Un risultato simile è stato ottenuto in [10] in cui è proposto un modello basato su multigrafo diretto e risolto tramite un algoritmo euristico di rilassamento, il quale cerca la soluzione ottimale del problema della programmazione ferroviaria (*railway timetabling problem*, RTP) rispettando dati vincoli topologici ed operativi. Questi modelli sono in ogni caso deterministici, dal momento che essi si basano su insiemi di dati pre-calcolati e non prendono in considerazione, durante il processo di ottimizzazione, i possibili effetti delle perturbazioni della circolazione.

Lo studio della circolazione perturbata è un argomento di primo piano nella ricerca ferroviaria: come discusso estensivamente in [5] (capitoli 10, 12 e 13) esso può essere efficacemente affrontato ricorrendo a metodi simulativi, che dal canto loro richiedono l'implementazione di algoritmi di riprogrammazione. Quest'ultimi di fatto simulano il compito dei centri di controllo della circolazione nell'effettuare le operazioni di riarrangiamento nello spazio e nel tempo dei tracce orario (*re-routing, re-timing, re-ordering*) al fine di ridurre i ritardi secondari.

analysis, some significant basic assumptions are still present.

These approaches introduce the problem of evaluating the required ranges or statistical distributions. This task can be accomplished by analysing a number of timetables, which can either be real ones or feasible virtual ones (e.g. constructed in order to analyse future situations), the results of which led to the first draft of the method presented in this paper.

This task involves optimisation issues which must be introduced in order to obtain timetables that are not only feasible (without conflicts) but which also meet the desired quality parameters. This problem, which has been studied by various researchers through different approaches, requires rather sophisticated mathematical tools, and has to be treated in a proper way. A compendium of the relevant state-of-the-art papers can be found in [5] where several timetabling and optimisation models have been proposed and discussed in chapters 8 and 9.

Within this research field, the application of a genetic algorithm (GA) was first considered in an attempt to solve the automatic timetabling problem, according to the indications of several literature sources. For example, a timetable generation methodology, based on a genetic algorithm, was proposed in [6], in which the mathematical issues that characterise the matter are presented comprehensively. Different approaches to solving these issues are also discussed and compared, thus demonstrating the effectiveness of the proposed model. Similarly, genetic algorithms were used in [7] to solve a railway timetable problem related to the allocation of train routes within a complex network, but without referring to the arrangement of the trains along a single line, thus highlighting how it is necessary to introduce heuristic ratios to improve the algorithm's performance. Analogous conclusions were attained in [8].

A heuristic algorithm integrated in a calculation tool (MOM system) that is able to calculate optimal periodic timetables, is presented in [9], after a discussion on the available railway simulation packages. The algorithm is therefore generalised to non-periodic timetables. A similar result was obtained in [10], where a model based on a direct multigraph was proposed and then solved through a relaxation heuristic algorithm, which searches for an optimal solution of the timetabling problem, while respecting given topological and operational constraints. However, these models are all deterministic, since they are based upon pre-calculated sets of data and do not take into consideration the possible effects of perturbations during the optimisation process.

The study of perturbed traffic is an important topic in railway research: as extensively discussed in [5] (chapters 10, 12 and 13), it could effectively be approached by making use of simulation methods which require the implementation of rescheduling algorithms. These actually model the dispatching centre tasks of re-timing, re-ordering or

In [11] gli autori trattano il problema della riprogrammazione in presenza di perturbazioni, determinando una serie di regole di priorità integrate in un sistema di decisione automatico, in grado di riprogrammare un orario perturbato. In [12] questi concetti vengono ulteriormente sviluppati, presentando un algoritmo altamente sofisticato che risolve iterativamente un problema macroscopico di gestione dei ritardi interfacciandosi al tempo stesso con un modello microscopico di simulazione della marcia dei treni. In questo modo viene effettuata un'azione di riprogrammazione ottimizzata, in grado di gestire la circolazione su un'intera rete e non unicamente su una singola linea.

In [13] un modello basato sulle reti di Petri simula il funzionamento di tutti gli elementi tecnici e logici che gestiscono la circolazione ferroviaria. Il modello è in grado di simulare la circolazione in condizioni sia nominali che perturbate, facendo uso di semplici criteri di riprogrammazione, i quali sono in grado di inserire precedenze non programmate di treni in ritardo se necessarie o analoghe misure operative.

In [14] viene sviluppata una metodologia per la generazione di orari, la quale considera esplicitamente gli effetti delle perturbazioni tramite un algoritmo di riprogrammazione incorporato. Gli orari così costruiti sono ottimizzati rispetto a parametri sia deterministiche che stocastiche. Nello stesso articolo sono inoltre presentati diversi criteri di riprogrammazione - con complessità ed efficacia crescenti - e le loro prestazioni sono valutate con diversi tipi di scenari di perturbazione.

In [15] l'intero RTP è affrontato sistematicamente, proponendo un ambiente di simulazione in grado di generare orari che soddisfino parametri di qualità prefissati, trattati come indicatori chiave di prestazione (*Key Performance Index*, KPI) per questo argomento. Lo studio estende ulteriormente le considerazioni proposte in [5] (capitolo 14) ed in [16] a proposito dei criteri e dei parametri adatti per misurare la qualità degli orari, i quali possono essere riassunti come proposto in tabella 1.

In particolare le prime tre voci riguardano il campo della *progettazione dell'orario*, dal momento che essi possono essere visti come indicatori della qualità di un orario in sé. D'altro canto, la stabilità e la robustezza sono valori utili per esaminare il comportamento di un orario durante l'esercizio. Lo strumento proposto in [15] è in grado di considerare contemporaneamente questi due campi principali, grazie a tre livelli distinti e complementari - macroscopico, microscopico e fine-tuning - che risolvono il problema dell'orario secondo i principi dell'ottimizzazione multi-obiettivo.

I KPI d'orario precedentemente menzionati possono essere usati anche durante valutazioni di capacità basate sugli orari (si veda oltre la sezione 4.1): in questo modo, per esempio, eventuali miglioramenti di una certa infrastruttura possono essere analizzati attraverso l'aumento delle performance degli orari realizzabili su di essa.

(locally) re-routing perturbed trains, in order to reduce secondary delays.

In [11], the authors dealt with the rescheduling problem for the case of perturbations, determining a series of priority dispatching rules that could be integrated with an automated decision support system, and which could be suitable for re-arranging a perturbed timetable. These concepts were further developed in [12], in which a highly sophisticated algorithm was provided. This algorithm iteratively solves a macroscopic delay management problem, and at the same time deals with a microscopic train-scheduling model. An optimised rescheduling, suitable for effectively managing traffic over a whole network system, and not only on a single line, was thus provided.

The model based on Petri nets presented in [13] allows the operations of all the elementary technical and logical elements that manage railway traffic to be simulated. The model allows traffic to be simulated in both scheduled and perturbed conditions, and makes use of simple rescheduling rules that are able to introduce an un-scheduled overtaking of delayed trains, when necessary, or similar operational measures.

A methodology for the generation of timetables has been developed in [14]; it explicitly considers, through an embedded rescheduling algorithm, the effects of traffic perturbations: the provided timetables are optimised with regard to both deterministic and stochastic parameters. Different rescheduling ratios - with increasing complexity and effectiveness - are also presented in this paper, and their performances are evaluated with different types of perturbation scenarios.

The whole railway timetabling problem has been treated comprehensively in [15], and a scheduling environment that can provide timetables which satisfy given quality parameters, treated as Key Performance Indexes (KPI) for the timetabling problem, has been proposed. The study further extends the considerations proposed in [5] (chapter 14) and in [16] pertaining to the criteria and the parameters suitable for measuring timetable quality, as summarised in table 1.

The first three items pertain to the timetable design field, since they can be considered as indicators of the quality of a timetable itself. On the other hand, timetable stability and robustness can be considered useful values to assess the performance of a timetable during operations. The tool proposed in [15] allows these two main aspects to be considered at the same time, thanks to the presence of a three-level environment (macroscopic, microscopic and "fine tuning") which solves the timetabling problem as a multi-objective optimisation.

The abovementioned timetable KPIs can also be used during timetable-based capacity evaluations (see section 4.1.): for example, infrastructure improvements can be evaluated through a quality enhancement of the relevant feasible timetables.

TABELLA 1 – TABLE 1

KPI per gli orari ferroviari, come proposto in [15] e [16]
Key Performance Indexes for railway timetables, as proposed in [15] and [16]

Tempo di viaggio programmato <i>Scheduled travel time</i>	Tempo complessivo di viaggio, composto dai minimi tempi di percorrenza, dagli allungamenti di percorrenza, dai minimi tempi di sosta e dai tempi di sosta addizionali dovuti ad incroci e precedenze <i>Travel time, composed of the minimum run times, the run time extensions, the minimum dwell times and the additional dwell times due to crossings or overtaking</i>
Occupazione dell'infrastruttura <i>Infrastructure occupation</i>	Calcolata secondo la fiche UIC 406 R [17] <i>Calculated accordingly to UIC code 406 [17]</i>
Realizzabilità dell'orario <i>Timetable feasibility</i>	Capacità dei treni di rispettare effettivamente il rispettivo orario nella realtà. Un orario realizzabile è pertanto privo di conflitti programmati <i>Ability of all trains to adhere to their actual schedule. A feasible timetable is in fact conflict free</i>
Stabilità dell'orario <i>Timetable stability</i>	Capacità dell'orario di assorbire i ritardi primari evitando la diffusione di ritardi secondari senza modifiche dello schema dell'orario (cioè senza un riarrangiamento attivo delle tracce) <i>Ability of the timetable to absorb primary delays, thus avoiding the spread of secondary delays, without modifying the timetable pattern (i.e. without any active dispatching)</i>
Robustezza dell'orario <i>Timetable robustness</i>	Capacità dell'orario di tollerare errori di progettazione o variazioni nei parametri operative, minimizzando l'insorgenza di ritardi primari <i>Ability of the timetable to withstand design errors or variations in the operational parameters, thus minimising the occurrence of primary delays</i>
Resilienza dell'orario <i>Timetable resilience</i>	Capacità dell'orario di ridurre i ritardi primari e secondari tramite la riprogrammazione delle tracce <i>Ability of the timetable to reduce primary and secondary delays through dispatching</i>
Consumo energetico <i>Energy consumption</i>	Energia consumata dai treni presenti nell'orario <i>Energy consumed by the traffic</i>

2. L'ambiente di simulazione

2.1. Descrizione generale

L'ambiente di simulazione si sviluppa su due livelli, uno *microscopico* ed uno *mesoscopico*. Entrambi si basano sullo schema concettuale dell'infrastruttura ferroviaria proposto nella fiche UIC 406 R [17], che fornisce alcune linee guida per il calcolo e l'analisi della capacità delle linee. Qualora si analizzi una rete ferroviaria, è possibile suddividerla in *linee di calcolo*, che possono essere viste come gli archi di un grafo (la rete ferroviaria, in cui i nodi sono rappresentati dalla maggiori stazioni). Senza prendere in considerazione analisi sviluppate a livello di grafo (tipiche della pianificazione dei sistemi di trasporto), è possibile effettuare valutazioni di capacità considerando un'unica linea di calcolo alla volta, definendola dunque come l'unità fondamentale. Una linea di calcolo può essere poi suddivisa in sezioni di calcolo (*calculation sections*, CS) le quali, secondo la fiche UIC 406, sono tratti di linea all'interno dei quali la composizione e la sequenza del traffico ferroviario non può cambiare, cioè dove incroci e precedenze non possono aver luogo. Pertanto, ciascuna sezione di calcolo è delimitata da due aree di stazione, in cui deviatoi e binari secondari permettono di alterare la sequenza dei treni. Infine, ciascuna CS è composta da sezioni di blocco, le quali condizionano il minimo distanziamento temporale ammesso tra treni successivi.

Entrambi i livelli simulativi consistono in primo luogo in un ambiente dedicato alla creazione degli orari, il

2. The simulation environment

2.1. General description

The simulation environment has been developed over two layers: a microscopic and a mesoscopic one. Both of these layers are based on the conceptual scheme of the railway infrastructure proposed in UIC leaflet 406 R [17], which sketches some guidelines for capacity calculations and assessments. When analysing a railway network, it is possible to subdivide it into calculation lines, which can be considered as the edges of a graph (the rail network). Without taking into account analyses developed at a graph level (which is typical of the transport-planning field), it is possible to carry out capacity assessments considering only one calculation line at a time, which can be seen as the major unit. A calculation line can be split into calculation sections (CS) which, according to the UIC 406 leaflet, are line stretches in which the traffic composition and sequence cannot change, that is, where no crossing or overtaking can take place. Therefore, each calculation section is delimited by two station areas, where turnouts and sidings permit the sequence of trains to be altered. Finally, each calculation section is composed of block sections, which condition the minimum admitted headway between consecutive trains.

Both simulation layers consist of a timetable-editing environment, which permits a timetable to be created and trains to be entered (and deleted), by specifying their arrival time in the first encountered station and their dwell time in all the stations along their route. With this information, to-

quale permette di modificarli inserendo (o eliminando) tracce specificando l'istante in cui i treni entrano nella *calculation line* in esame - l'orario di arrivo nella prima stazione ad essere incontrata - e il loro tempo di sosta in ciascuna stazione lungo il percorso. Con queste informazioni, insieme ai dati tecnici descritti nel seguito, il simulatore è in grado di calcolare l'intera traccia orario lungo la linea di calcolo.

È inoltre possibile generare tracce che non potrebbero essere effettuate nella realtà a causa di conflitti con altre tracce, a loro volta protette dal sistema di segnalamento. Per queste ragioni in entrambi i livelli di simulazione è implementata una sub-routine che cerca e segnala all'utente - o all'algoritmo di programmazione, come spiegato più avanti - la presenza di conflitti di circolazione nell'orario programmato.

2.2. Livello di simulazione microscopico

L'ambiente di simulazione microscopico è in grado di calcolare i diagrammi di marcia (cioè i diagrammi spazio-tempo o spazio-velocità) dei treni usando come dati di input le caratteristiche tecniche di infrastruttura, sistema di segnalamento e materiale rotabile:

- i raggi ed i gradienti rispettivamente di curve e livellate presenti lungo la linea, insieme con la loro posizione. Come alternativa, è possibile inserire direttamente il grado di prestazione lungo la linea;
- i limiti di velocità per le due direzioni di marcia, che possono essere definiti per cinque diversi ranghi di velocità;
- la posizione e le caratteristiche delle stazioni, cioè numero e lunghezza dei binari di sosta, eventuali limitazioni di velocità per ciascun itinerario, posizione dei segnali di protezione e di partenza. Le stazioni sono modellizzate considerando un layout semplificato e convenzionale, riportato in fig. 1;
- il materiale rotabile è modellizzato tramite la massa, la lunghezza e la prestazione dei treni. La prestazione in trazione è inserita tramite il diagramma forza di trazione-velocità, insieme alla massima velocità caratteristica di ciascun treno ed alla formula trinomiale delle resistenze ordinarie al moto, definita tramite i coefficienti. In maniera analoga ad altri programmi di simulazione commerciali, nonché alla logica dei Computer Vitali di bordo treno (per esempio in ERTMS), la frenatura è modellizzata a decelerazione costante attraverso una *decelerazione standard*.

Con questi dati, insieme alle "condizioni al contorno" riguardanti l'orario di arrivo nella prima stazione ed i tempi di sosta ciascuna di esse, il simulatore calcola i profili di marcia dei treni tramite l'integrazione numerica dell'equazione fondamentale del moto. Di fatto, il simulatore in primo luogo genera un profilo di velocità "dinamico" a partire da quello "statico" definito dai limiti di velo-

gether with the described technical data, the simulator allows the whole train path along the line to be calculated.

It is possible to edit the train paths that in reality could not be run due to conflicts with other paths protected by the signalling system. For these reasons, a sub-routine is implemented in both simulation levels; this routine looks for and points out the presence of conflicts in the scheduled timetable to the user (or to the scheduling algorithm, as explained later on).

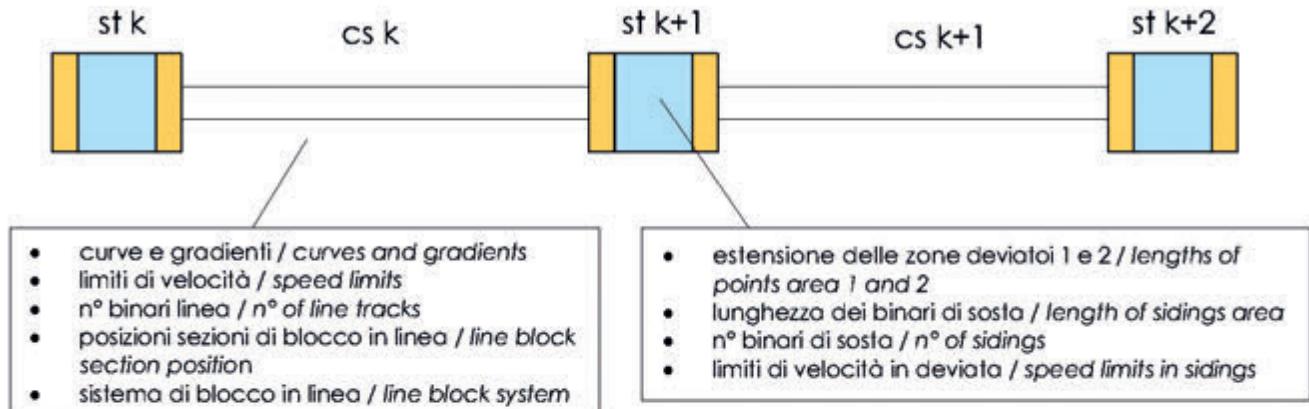
2.2. Microscopic simulation level

The microscopic simulation environment allows the travel diagrams (i.e. space-time or space-speed diagrams) of trains to be calculated using the technical features of the infrastructure, the signalling system and the rolling stock as input data:

- *the radii and gradients of the curves and slopes present along the line, together with their positions. As an alternative, it is possible to directly enter the performance index values along the line;*
- *the speed limits for the two running directions, which can be defined for up to five different speed classes;*
- *the position and features of the stations, i.e. the number of tracks, possible speed limitations for secondary sidings, their lengths and the position of the home and departure signals. The stations are modelled considering a simplified and conventional layout, as shown in fig. 1;*
- *the rolling stock is modelled considering the mass, length and performance of the trains. The tractive performance is entered through the traction forces vs. speed diagram, together with the maximum speed (characteristic of each train) and the ordinary motion resistance formula, which is defined by trinomial coefficients. As in other commercial packages, as well as in the logics of Vital Computers (e.g. in ERTMS), braking is modelled as constant deceleration through a standard deceleration rate.*

With these data, together with the "boundary conditions" about the arrival time at the first station and the dwell times in all the stations, the simulation core calculates the travel profile of trains through a numerical integration (along the space) of the standard motion equation. The simulator in fact first generates a "dynamic" speed limitation profile, starting from the "static" one defined by the infrastructural speed limits. This dynamic profile introduces the braking curves (conditioned by the standard deceleration rate) that the train driver follows to reduce speed. Subsequently, considering the dynamic profile as an upper bound, the actual speed profile is obtained through a numerical integration of the motion equation. The integration is performed between consecutive train stops. Stops can only be scheduled at stations by setting a dwell time greater than zero. The following outputs are provided:

Modello di linea / Line model



Modello di stazione / Station model

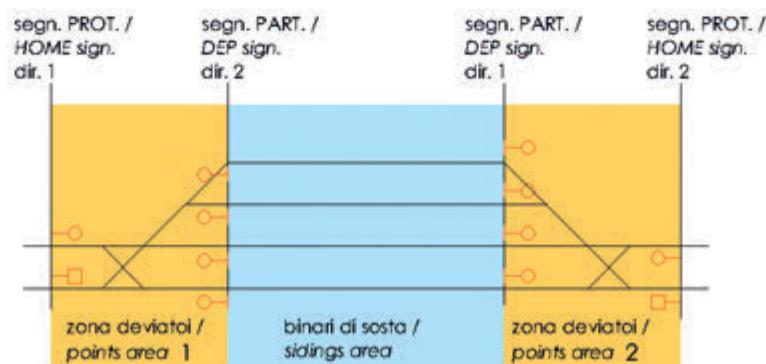


Fig. 1 - Modello di linea e stazioni nell'ambiente di micro simulazione.
Fig. 1 - Model of the line and stations in the micro-simulation environment.

cità della linea. Questo profilo dinamico introduce le curve di frenatura - condizionate dalla decelerazione standard - che il treno deve seguire per ridurre la propria velocità. Successivamente, considerando il profilo dinamico come un limite superiore, il profilo effettivo di velocità è ottenuto attraverso l'integrazione numerica dell'equazione del moto. L'integrazione è effettuata fra ogni coppia di fermate consecutive. Le fermate possono essere impostate solo nelle stazioni imponendo un tempo di sosta maggiore di zero. Vengono forniti i seguenti output:

- diagrammi di marcia, disponibili come grafici spazio-tempo e spazio-velocità;
- tempi di viaggio tra le stazioni;
- diagrammi spazio-sforzo di trazione, che possono essere usati per calcolare i consumi energetici ai cerchioni.

I diagrammi di marcia sono inoltre adoperati per calcolare l'occupazione delle sezioni di blocco, la quale permette di definire la cosiddetta *block occupation stairway* [5], [17] e di identificare i possibili conflitti di circolazio-

- travel diagrams, available as space-time and space-speed graphs;
- travel times between stations;
- space-tractive effort diagrams, which can be used to calculate the energy consumptions at the wheel rims.

The travel diagrams are also employed to calculate the block section occupations, which in turn permits the so-called block occupation stairway to be defined [5], [17] and any possible traffic conflict to be identified and located. Line block sections are defined as input data, while station block sections are defined according to the simplified station model displayed in fig. 1: each siding in a station constitutes a block section, while the two "points areas" represent two additional block sections subject to special occupation or clearing rules, depending on the type of block system that is modelled.

Once this computation has been conducted for each scheduled train, the conflict detection subroutine is run. This routine checks, in the timetable, whether more than one train simultaneously uses a block section, by verifying

ne. Le sezioni di blocco di linea sono definite come dati di input, mentre le sezioni di stazione sono definite secondo il modello di nodo presentato in fig. 1: ciascun binario di sosta costituisce una sezione di blocco mentre le due "zone deviatoi" rappresentano due sezioni aggiuntive soggette a speciali regole di occupazione e rilascio, a seconda del tipo di sistema di blocco modellizzato.

Questi calcoli vengono svolti per ciascun treno in modo tale da poter lanciare il sottoprogramma di ricerca dei conflitti di circolazione. Esso controlla nell'intero orario se più di un treno occupa contemporaneamente la stessa sezione di blocco, verificando se sono presenti intersezioni tra gli intervalli di occupazione.

Questo modello di micro simulazione presenta la maggior limitazione nel layout, convenzionale e semplificato, utilizzato per le stazioni. Infatti, per avere un modello accurato e completo, un nodo ferroviario non può essere rappresentato diversamente da un grafo che rispecchi la sua reale topologia. Le limitazioni introdotte da questo modello di stazione semplificato possono essere riassunte come segue:

- i binari di stazione sono tutti caratterizzati dalla stessa lunghezza;
- i treni si fermano in stazione al segnale di partenza. Questo può essere fonte di inaccuratezza soprattutto quando si considerano grandi stazioni dove i segnali di partenza si trovano ben oltre il limite delle banchine;
- le zone deviatoi sono considerate come un'unica sezione di blocco, una sorta di "scatola nera" al cui interno si trascura la topologia del nodo (in altre parole, non si tiene conto delle matrici degli itinerari di stazione). Per quanto riguarda la localizzazione dei conflitti, la presenza contemporanea di due treni è permessa se essi provengono da differenti binari di linea e se procedono verso diversi binari di stazione (e vice-versa). Si assume che questa semplificazione sia compatibile con gli scopi per cui è stato sviluppato il modello.

Al di là di questi limiti, lo strumento di microsimulazione ha presentato una buona accuratezza nel calcolo dei tempi di percorrenza e nei diagrammi di marcia, se confrontato sia con orari reali sia con gli output di programmi di simulazione ferroviaria commerciali (OpenTrack). In ogni caso, esso fa uso di una significativa quantità di dati di ingresso, ed il processo di calcolo di un intero orario - l'integrazione numerica dell'equazione del moto per ciascun treno - ha tempi computazionali piuttosto lunghi, dell'ordine delle decine di secondi, a seconda della lunghezza della linea e del numero di treni. Pertanto un tale approccio di calcolo non è adatto per implementare un algoritmo di programmazione automatico, in cui i sottoprogrammi di calcolo dell'orario e di ricerca dei conflitti vengono lanciate centinaia se non migliaia di volte. Per questi motivi è stato sviluppato un modello mesoscopico, più adatto a raggiungere gli obiettivi prefissati.

whether any intersection is present in the occupation time slots.

This microsimulation model features the greatest limitations in the simplified and conventional station layout. In fact, for an accurate and complete model, a railway node cannot be represented in a different way from a graph, traced following its real topology. The limitations introduced by this simplified station model can be summed up as follows:

- the station sidings all have the same length;*
- trains stop in a station at the departure signal position. This is mainly an issue when large stations are considered in which the departure signals are placed far beyond the platform limits;*
- the points areas are considered as one whole block section, that is, a "black box" in which the node topology is neglected (i.e. no station route matrix is defined or taken into account). As regards the detection of conflicts, the simultaneous presence of two trains is admitted, if they come from different line tracks and they proceed to different station tracks, and vice versa. This simplification has been assumed to be compatible with the aims of this tool.*

Apart from these limitations, the microsimulation tool has proved to be accurate for the calculation of running times and travel diagrams, compared with both real timetables and the outputs of commercial railway simulation packages (OpenTrack). Moreover, it makes use of a significant amount of input data; the calculation process of a whole timetable (the numerical integration of the motion equation for each train) has a rather long computation time, of the order of some tens of seconds, depending on the length of the calculation line and on the number of trains that are present. Therefore, this calculation core is not suitable for implementing an automatic scheduling algorithm, where the timetable calculation and conflict detection sub-routines would be called upon hundreds or thousands of times. This is why a mesoscopic model has been developed.

2.3. Mesoscopic simulation level

The mesoscopic environment has specifically been conceived and developed to implement automatic timetabling functions. Therefore, at this level, the computational efforts should be reserved for the scheduling algorithm itself and should not be scattered over other tasks, such as the periodic and frequent re-calculations of each train path. The idea is therefore to assume that trains can be categorised into classes that are internally homogeneous according to their speed and priority. It is worth noticing that such an assumption consistently matches the actual train classification commonly adopted by railway operators.

Therefore, it is possible to avoid a numerical calculation of all the travel diagrams by characterising each class

2.3. Livello di simulazione mesoscopico

L'ambiente mesoscopico è stato specificatamente concepito e sviluppato per implementare funzioni di *automatic timetabling*. Di conseguenza, in questo livello gli sforzi computazionali devono essere riservati per l'algoritmo di programmazione e non possono essere dispersi in altri compiti, come il periodico e frequente calcolo di ciascuna traccia. L'idea è quindi di assumere che i treni possano essere suddivisi in classi, le quali sono internamente omogenee per velocità e priorità. È opportuno evidenziare come tale assunzione sia coerente con l'effettiva classificazione dei treni comunemente impiegata dagli operatori ferroviari.

Di conseguenza è possibile evitare il calcolo di tutti i diagrammi di marcia caratterizzando ciascuna classe tramite alcune quantità temporali. Quest'ultime, descritte in fig. 2, sono sufficienti per descrivere ciascun treno in un ambiente esplicitamente dedicato al *timetabling* come quello mesoscopico.

In ciascuna sezione di calcolo (CS), per ogni classe di treni *cl* e per entrambe le direzioni di marcia:

- rt è il tempo di percorrenza necessario per attraversare la CS, calcolato per un treno che non fermi né nella stazione di partenza né in quella di arrivo;
- t_{ad1} è il tempo di percorrenza addizionale (rispetto ad rt) per un treno che ferma nella stazione di partenza, dovuto al transitorio di accelerazione;
- t_{ad2} è il tempo di percorrenza addizionale per un treno che ferma nella stazione di arrivo, dovuto al transitorio di decelerazione.

L'occupazione delle sezioni di calcolo è definita come il tempo in cui un treno occupa il binario singolo tra due stazioni consecutive, impedendo ogni partenza dalla stazione successiva nella direzione di marcia a lui opposta. Questa definizione è coerente col criterio con cui sono localizzati i conflitti tra treni che mariano in direzioni opposte su sezioni a binario singolo, e non deve essere confusa con l'occupazione delle sezioni di blocco (il *blocking time* secondo la definizione data in [17] o in [5], capitolo 2). Quest'ultimo non appare esplicitamente nel modello mesoscopico (fig. 3).

Nel modello mesoscopico l'occupazione di una CS è calcolata a partire dagli istanti di partenza ed arrivo nelle stazioni che la delimitano. A tal fine sono necessari due ulteriori parametri:

- t_{o1} è il tempo che trascorre tra l'inizio dell'occupazione della CS e la partenza del treno;

with temporal quantities. The quantities, described in fig. 2, are sufficient to describe each train in such a timetabling environment as the mesoscopic one.

In each calculation section (CS), for each train class and for both the running directions:

- rt is the running time necessary to cross the CS, calculated for a train which does not stop in the departure station or in the arrival one;
- t_{ad1} is the additional running time (with respect to rt) for a train which stops in the departure station, due to the acceleration transient;
- t_{ad2} is the additional running time for a train which stops in the arrival station, due to the deceleration transient.

The occupation of the calculation sections is defined as the time in which the train occupies the single track between two consecutive stations, thus preventing any departure in the opposite direction from the next station. This definition is consistent with the way in which conflicts between opposite trains are detected on single-track sections, and should not be confused with the occupation of the block sections (the blocking time as defined in [17] or [5], chapter 2). The latter does not figure explicitly in the mesoscopic environment (fig. 3).

In the mesoscopic model, the CS occupation is calculated considering the departure and arrival times in the stations bordering the calculation section. For this purpose, two other parameters are necessary:

- t_{o1} is the time that has elapsed between the beginning of the occupation of the CS and the departure of a train;
- t_{o2} is the time that has elapsed between the arrival of a train and the end of the occupation of the CS.

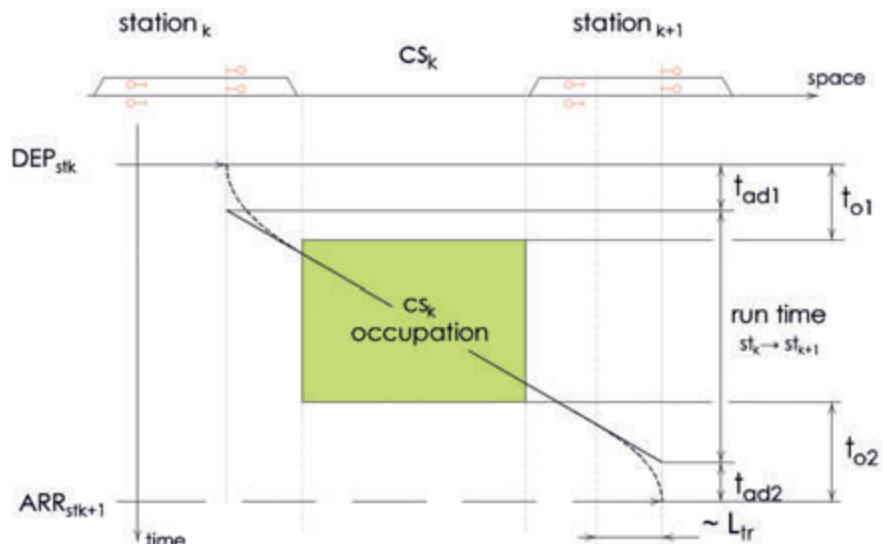


Fig. 2 - Parametri impiegati nel modello mesoscopico.

Fig. 2 - Parameters adopted in the mesoscopic model.

- t_{o2} è il tempo che trascorre tra l'arrivo del treno e la fine della sua occupazione della CS.

Per ciascuna classe e CS, t_{o1} e t_{o2} possono assumere diversi valori a seconda che il treno fermi o meno nelle stazioni di estremità. Infatti, come illustrato da fig. 3, l'occupazione può cambiare significativamente a causa di questa "condizione al contorno". Questo è dovuto al diverso profilo del diagramma di marcia e al fatto che nel caso di fig. 4 è necessario considerare anche il tempo di approssimazione relativo al segnale di partenza della prima stazione.

Se l'estensione della stazione è significativamente più lunga della lunghezza dei treni, prendendo così la forma di un tratto di doppio binario parziale (fig. 4b), il tempo

t_{o1} and t_{o2} can assume different values for the same class and CS, depending on whether the train stops or does not stop in the starting and finishing stations. In fact, as can be seen in fig. 3, the occupation can change significantly according to this "boundary condition". This is due to the different shapes of the train-run diagram, and because, in the case of fig. 3a, it is also necessary to consider the approach time to the departure signals of the first station.

If the station limits are much longer than the train, thus taking the shape of a partial double-track section (fig. 4b), the CS occupation time (i.e. that related to the actual single-track stretch) can be reduced, and the t_{o1} and t_{o2} parameters can assume negative values (the CS occupation could begin after the departure of the train).

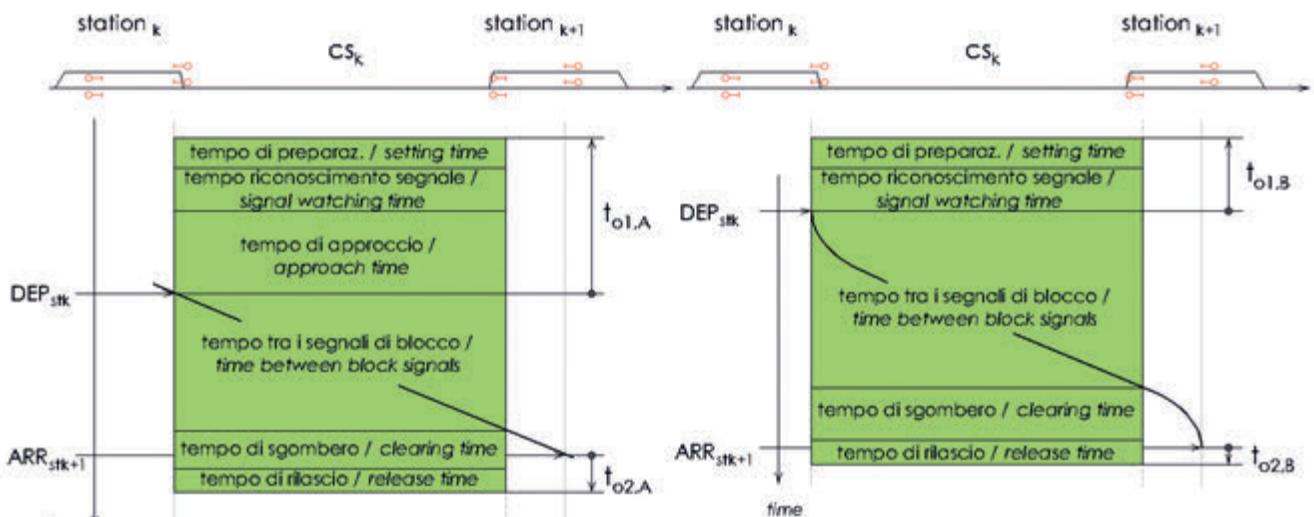


Fig. 3 - Occupazione di una sezione di calcolo (aree verde chiaro) nel caso di un treno che ferma (b) o meno (a) nelle stazioni di estremità [5].

Fig. 3 - Occupation of a calculation section (light green area) in the case of a train stopping (b) or not stopping (a) in the starting and finishing stations [5].

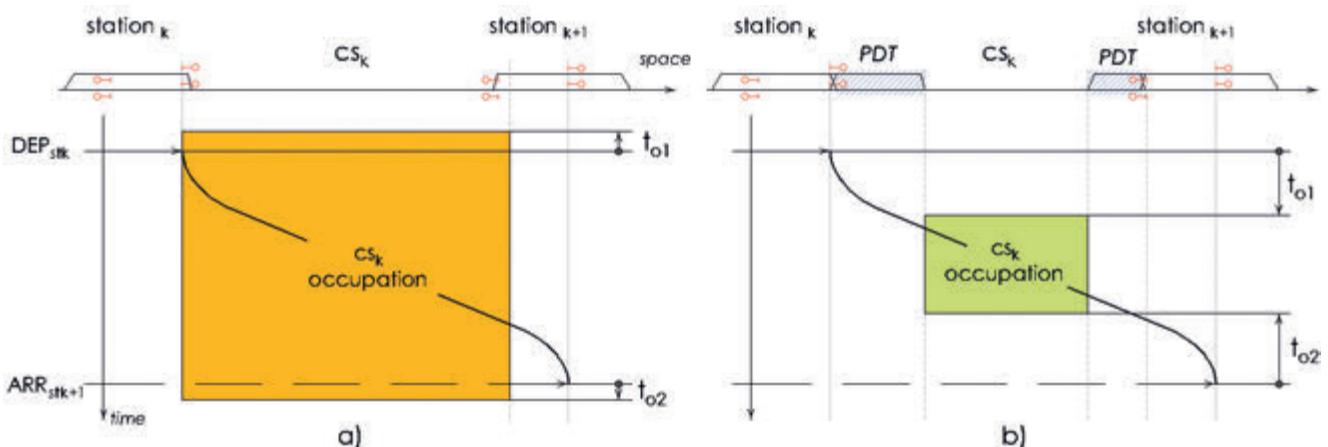


Fig. 4 - Tempo di occupazione di una CS nel caso di assenza (a) e presenza (b) di doppio binario parziale (PDT, partial double track).

Fig. 4 - CS occupation time in the case of the absence (a) and presence (b) of a partial double track.

di occupazione di una CS - quello cioè relativo al tratto effettivamente a binario singolo - può essere ridotto, e i relativi parametri t_{o1} e t_{o2} possono assumere valori anche negativi (l'occupazione di una CS potrebbe cominciare dopo la partenza del treno).

Con questa definizione, l'occupazione della sezione di calcolo rappresenta di fatto il minimo distanziamento temporale ammesso tra due treni che viaggiano in direzioni opposte sullo stesso binario. È quindi necessario definire anche il minimo distanziamento h ammesso tra due partenze successive di treni che viaggiano nella stessa direzione. Al fine di tenere adeguatamente conto dell'eterotachicità esistente fra treni di classi diverse, h è definito per ogni possibile coppia di classi (oltre che per ciascuna sezione di calcolo e per entrambe le direzioni di marcia), considerando anche il loro ordine reciproco. Inoltre, allo stesso modo di t_{o1} e t_{o2} , h è influenzato dalle velocità iniziale e finale che i treni hanno in ciascuna sezione di calcolo, dal fatto cioè che essi fermino o meno nelle stazioni di estremità. Un singolo treno può pertanto sottostare a 4 diverse "condizioni al contorno" e qualora si consideri una coppia di treni le possibili configurazioni salgono a 4^2 .

Si può notare come ciascun insieme di informazioni (dataset) sia composto da un significativo numero di dati (tabella 2). Pertanto, anche se è possibile caricare manualmente un intero dataset, è stata implementata una procedura facilitata, la quale permette di ottenere facilmente un dataset dall'ambiente di microsimulazione. Un algoritmo dedicato simula automaticamente le corse dei treni nelle configurazioni richieste, al fine di ricavare e, qualora necessario, estrapolare (per esempio, nel caso del calcolo del minimo distanziamento ammesso in ciascuna CS) i dati interessanti. Pertanto, da questo punto di vista, la vera utilità del simulatore microscopico è ricavare dataset completi, pronti per essere impiegati nell'ambiente mesoscopico.

According to this definition, the occupation of the calculation section actually represents the headway between two trains running on the same track but in opposite directions. It is therefore necessary to define the minimum admitted headway, between two consecutive departures of trains running in the same direction. In order to take into account the speed heterogeneity of different train classes, h is defined for each possible couple of classes (but also for each calculation section and for both running directions), considering their reciprocal order. Moreover, in the same way as for t_{o1} and t_{o2} , h is affected by the initial and final speeds that trains have in each calculation section, i.e. whether they stop or do not stop in the starting and ending stations. A single train can therefore have 4 different "boundary conditions", and when a couple of them are considered, the possible configurations rises to 4^2 .

It is possible to notice how each dataset is actually composed of a significant amount of data (table 2). Even though it is possible to manually enter a whole dataset, a smart procedure, which permits a dataset to be obtained easily from the microsimulation environment, has been implemented. A proper algorithm automatically simulates the train runs in the required configurations, in order to provide and, when necessary, extrapolate the relevant data (for example, in the case of computing the minimum admitted headways in each CS). Therefore, from this point of view, the real usefulness of the microscopic simulator is to provide complete datasets that are ready to be managed by the mesoscopic tool.

Thanks to such pre-defined datasets, it is possible to significantly reduce the computation time required by the code to calculate the timetable of each train (table 3). Since each dataset is unique for each class, in order to simulate even small differences in the running times between trains of the same class, it is necessary to define a parameter K_r for each

TABELLA 2 – TABLE 2

Parametri del dataset mesoscopico
Mesoscopic dataset parameters

Parametro Parameter	Significato Significance	Forma matriciale matematica e significato delle dimensioni Mathematical matrix form and dimension significance
rt	Tempo di percorrenza <i>Running time</i>	(classe · direzione · CS) (class · direction · CS)
t_{ad1}	Tempo di percorrenza aggiuntivo nella stazione di partenza <i>Additional running time in the departure station</i>	(classe · direzione · CS) (class · direction · CS)
t_{ad2}	Tempo di percorrenza aggiuntivo nella stazione di arrivo <i>Additional running time in the arrival station</i>	(classe · direzione · CS) (class · direction · CS)
t_{o1}	Anticipo dell'occupazione della CS nella stazione di partenza <i>CS occupation anticipation in the departure station</i>	(classe · direzione · CS · cond. cont. 1 ÷ 2) (class · direction · CS · bound. cond. 1 ÷ 2))
t_{o2}	Anticipo dell'occupazione della CS nella stazione di arrivo <i>CS occupation anticipation in the arrival station</i>	(classe · direzione · CS · cond. cont. 1 ÷ 2) (class · direction · CS · bound. cond. 1 ÷ 2))
h	Distanziamento minimo tra treni consecutivi <i>Headway between consecutive trains</i>	(classe · direzione · CS · cond. cont. 1 ÷ 16) (class · direction · CS · bound. cond. 1 ÷ 16)

Grazie a dataset definiti preliminarmente, si può drasticamente ridurre il tempo computazionale richiesto dal codice per calcolare l'orario di ciascun treno (tabella 3). Dal momento che un dataset è unico per ciascuna classe, per simulare differenze anche piccole di tempo di percorrenza tra treni della stessa classe è necessario definire, per ciascun treno e ciascuna sezione, un parametro K_{rt} . Tale parametro viene moltiplicato per il tempo di percorrenza al fine di modellizzare lo stiramento o il rilassamento delle tracce. Queste operazioni possono essere eseguite sia manualmente da parte dell'utente sia dall'algoritmo di programmazione, per esempio al fine di simulare un treno in ritardo che consuma gli allungamenti di percorrenza per recuperare il ritardo.

Usando come variabili di input l'orario di arrivo nella prima stazione ed i tempi di sosta in ciascuna stazione della linea, grazie al dataset precedentemente preparato il tempo computazionale impiegato per ciascun treno cala a pochi centesimi di secondo. Inoltre anche il sottoprogramma di ricerca dei conflitti risulta più semplice e veloce, dal momento che il confronto dei tempi di occupazione non viene più svolto a livello di sezione di blocco, ma bensì di *calculation section*. Una volta calcolato l'orario, è possibile ricavare gli intervalli di occupazione per ogni treno in ciascuna CS. Confrontando ogni coppia di intervalli consecutivi, si possono identificare le eventuali sovrapposizioni, le quali rappresentano conflitti tra treni che marciano in direzioni opposte. Tali conflitti sono ovviamente definiti solo in sezioni di calcolo caratterizzate dal singolo binario. Diversamente, i conflitti tra treni marcianti nella stessa direzione hanno luogo quando tra due partenze consecutive in una certa stazione non viene rispettato il distanziamento temporale definito nel dataset (per la CS e la coppia di classi interessate). Inoltre il sottoprogramma controlla, per entrambi i tipi di conflitto, se viene rispettato un dato tempo cuscinetto - stabilito dall'utente - generando poi un prospetto in cui ciascun conflitto presente viene elencato e descritto, riportando:

- i treni coinvolti;
- la sezione di calcolo in cui ha luogo il conflitto;
- l'istante in cui ha luogo il conflitto;
- il tipo di conflitto (direzione uguale od opposta);
- l'entità del conflitto (quanto si sovrappongono gli intervalli di occupazione).

L'ambiente di simulazione mesoscopico è implementato con un'interfaccia grafica, la quale permette di generare e modificare facilmente gli orari. Con riferimento alla fig. 5, gli oggetti e gli strumenti principali a disposizione dell'utente sono descritti di seguito:

train and for each calculation section. This parameter is multiplied by the running time necessary to model the stresses or relaxations of the train paths. This can be done both by the users during manual timetable editing or by the scheduling algorithm itself, for example to simulate a delayed train which uses up any run time allowance to recover delays.

If the arrival times in the first station and the dwell times in each station of the line are used as input variables, the computation time per train drops to a few hundredths of a second, because of the pre-arranged dataset. In addition, the conflict detection subroutine is simpler and faster, since the occupation time comparison is no longer carried out at a block section level, but at a calculation section one. Once the timetable has been calculated, it is possible to obtain the occupation intervals of all the trains for each CS. Comparing each couple of consecutive intervals, it is possible to identify the possible overlaps that represent conflicts between trains running in opposite directions. Such conflicts are obviously just defined for the calculation sections that are marked as single tracks. Nevertheless, conflicts between trains running in the same direction occur when the headway defined in the dataset (for the calculation section and couple of classes of interest), between two consecutive departures from a certain station, has not been respected. Furthermore, the subroutine checks, for both of the conflict types, whether a suitable buffer time (fixed by the user) has been respected, thus generating a register in which all the present conflicts are listed and described. This register reports:

- the trains that are involved;
- the calculation section in which the conflict takes place;
- the time at which the conflict takes place;
- the conflict type (opposite or same direction);
- the conflict entity (to what extent the occupation intervals overlap).

TABELLA 3 – TABLE 3

Pseudo-codice per il calcolo dell'orario, il quale impiega sia dati di input che un dataset

Pseudo-code for the timetable calculation, using input and dataset data

```
% INPUT data : ARR[1..] arrival time in the first station, DM, Krt
for itr = 1 to ntr
    for kst = 1 to nst-1
        for kst = 1 to nst-1
            % for all trains
            % for all stations
            Kst=kst
            DEP[kst,itr]=ARR[kst,itr]+DM[kst,itr]
            if DM[kst,itr] > 0
                A1=rt.ad1[class[itr],dir[itr],kcs]
            else
                A1=0
            end if
            if DM[kst+1,itr] > 0
                A2=rt.ad2[class[itr],dir[itr],kcs]
            else
                A2=0
            end if
            RT[kcs,itr]=rt[class[itr],kcs]+A1+A2
            ARR[kst+1,itr]=DEP[kst,itr]+RT[kcs,itr]*Krt[kcs,itr] % Krt is used for characterising
            % the path of each Train
        end for
    end for
end for
% OUTPUT data : ARR DEP DM which actually represent the whole timetable
```

- orario tabellare del treno selezionato, strumenti di editing;
- lista dei treni, per ciascuna classe;
- strumenti per importare/esportare i dataset memorizzati;
- orario grafico;
- lista dei conflitti tra treni marcianti in direzioni opposte;
- lista dei conflitti tra treni marcianti nella stessa direzione;
- tempo cuscinetto da rispettare;
- lista delle sezioni di calcolo a doppio binario;
- strumenti per il timetabling automatico.

È possibile gestire fino a tre classi, assunte come sufficienti per il carattere di ricerca di questo lavoro. Inoltre tale numero è adeguato a modellizzare una circolazione decisamente comune come quella composta, per esempio, da treni Regionali o locali, Intercity - o passeggeri veloci a lunga percorrenza - e Merci. Ad ogni modo in possibili sviluppi futuri potrà essere gestito un maggior numero di classi.

The mesoscopic environment is implemented with a graphical interface, which permits various timetable-editing operations to be carried out easily. As far as fig. 5 is concerned, the main available objects and tools are as follows.

- numerical timetable for the selected train, editing tools;
- trains lists for each class;
- stored dataset import/export tools;
- graphical timetable;
- list of conflicts between trains running in opposite directions;
- list of conflicts between trains running in the same direction;
- buffer time that has to be respected;
- list of double-track calculation sections;
- automatic timetabling tools.

It is possible to manage up to three classes, which has been assumed as sufficient for the research nature of this work. Furthermore, three classes are adequate to model a common traffic mix, such as that composed, for instance,

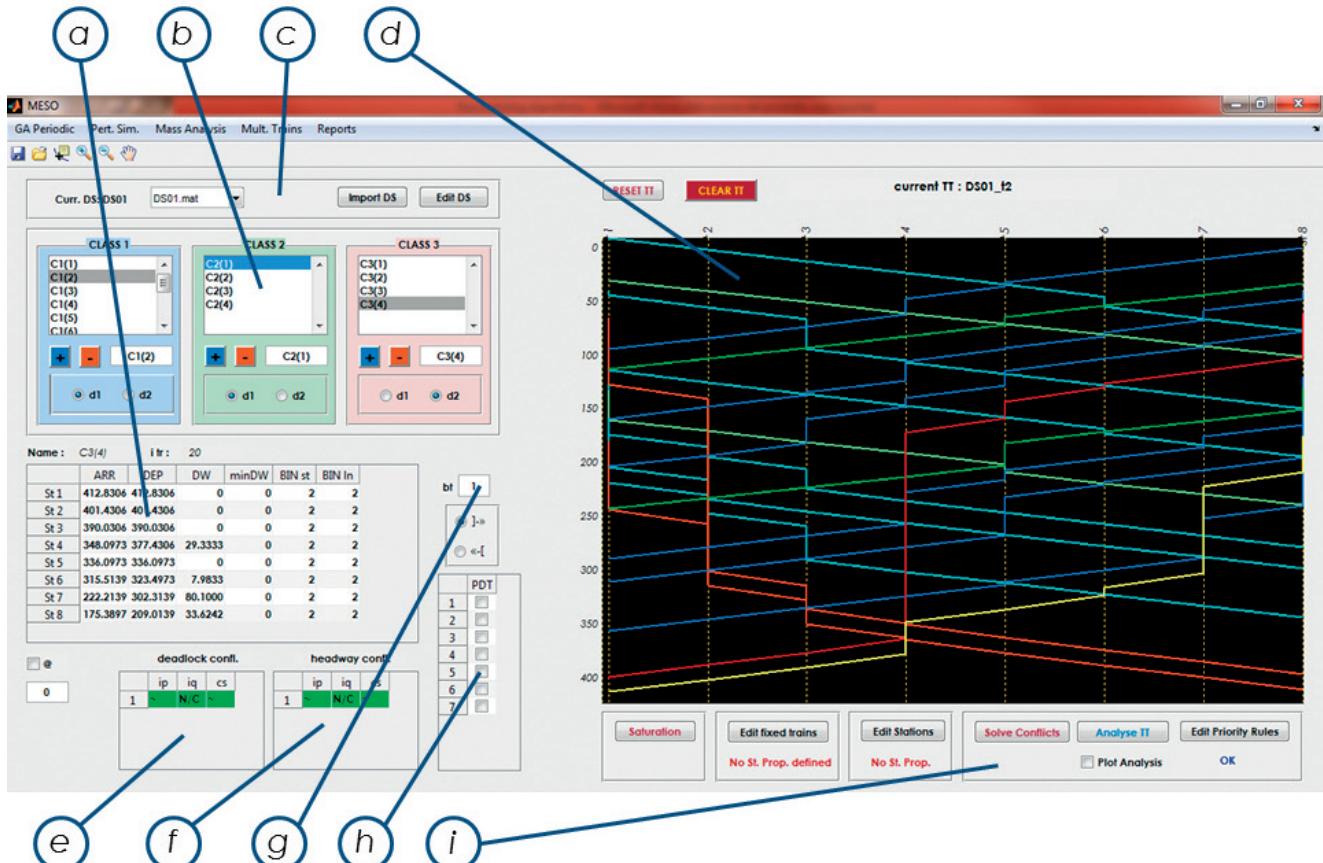


Fig. 5 - L'interfaccia grafica dell'ambiente di simulazione mesoscopico.
Fig. 5 - The graphical interface of the mesoscopic environment.

3. Algoritmi di programmazione

3.1. Un algoritmo genetico per la compilazione dell'orario

Un algoritmo di programmazione è una procedura in grado di modificare un orario sorgente al fine di risolvere tutti i conflitti di circolazione presenti, introducendo delle soste in stazione per incroci e precedenze. Un orario privo di conflitti potrebbe tuttavia essere realizzabile solo dal punto di vista teorico: di fatto, un algoritmo di programmazione deve restituire dei risultati che rispettino anche un certo numero di vincoli di qualità aggiuntivi (come per esempio il contenimento dei tempi di sosta aggiuntivi non strettamente richiesti dai servizi passeggeri o merci). In questo modo, quello della programmazione dell'orario (*timetabling*) viene ad assumere i connotati di un problema di ottimizzazione: i tempi di viaggio dei treni in orario devono essere minimizzati col vincolo di non avere conflitti.

L'algoritmo genetico (GA) è stato oggetto di un primo tentativo volto a risolvere questo problema. Tale algoritmo è particolarmente indicato per risolvere problemi non lineari di ottimizzazione vincolata, che presentino punti di minimo sparsi e separati da zone non ottime o i cui punti non rispettano i vincoli dati. Infatti tale algoritmo può esplorare lo spazio delle soluzioni senza rimanere intrappolato in punti di minimo locale. Il fenomeno studiato viene modellizzato tramite un vettore cromosoma (*chromosome array*), il quale è un vettore contenente un certo numero di variabili di controllo. Tali variabili devono soddisfare dati vincoli reciproci e sono adoperate per calcolare il valore della funzione di *fitness* (o di penalità). Tale funzione di fatto valuta lo scenario definito dal "seme" contenuto nel vettore cromosoma, secondo un dato criterio. Il seme di ciascuna iterazione è prodotto seguendo un criterio evoluzionario, sia facendo uso dei migliori cromosomi apparsi in precedenza sia tramite delle *mutazioni* casuali, le quali permettono di cambiare repentinamente la direzione di indagine dell'algoritmo.

Ai fini del *timetabling*, una volta stabilito quali e quanti treni n_{tr} debbano figurare in una data finestra temporale, il vettore cromosoma X risulta composto da $n_{tr} \cdot (n_{st} - 1)$ elementi, cioè dai tempi di sosta di ciascun treno in tutte le stazioni a parte le ultime.

Come spiegato precedentemente, questo seme - insieme ad un dataset precedentemente definito - è sufficiente per calcolare l'intero orario. Quest'ultimo può essere valutato per mezzo di una funzione di fitness definita come

$$f(X) = \sum_{cl=1}^3 (k_{cl} \cdot DW_{cl}) + c1 \cdot \Delta_{cfl}^{c2} \quad (1)$$

$$DW_{cl} = \sum_{tr \in cl} DW_{add,tr} = \sum_{tr \in cl} (DW_{tr} - DW_{min,tr}) \quad (2)$$

of Regional or local, Express (Intercity or fast long-haul passenger trains) and Freight trains. However, it is foreseen that a higher number of classes could be handled in future developments of the algorithm.

3. Scheduling algorithms

3.1. Genetic algorithm for timetable scheduling

A scheduling algorithm is a procedure that is able to modify a source timetable, in order to solve all traffic conflicts, by introducing dwell times in stations, for train crossings and overtaking. However, a conflict-free timetable is only feasible from a theoretical point of view: a scheduling algorithm should in fact provide results that are also compliant with a given number of additional quality constraints (e.g. to limit the additional dwell times not required by passenger or goods services). The timetabling problem thus takes the shape of an optimisation one: the travel times of the trains in the timetable should be minimised with the constraint of having no conflicts.

A first attempt to solve this problem resorted to a genetic algorithm (GA). Genetic algorithms are particularly useful for solving nonlinear constrained optimisation problems that feature scattered local minimum points separated by unfeasible or not-optimum zones, since it is able to explore the solution space without being trapped in local minimum points. The considered phenomenon is modelled by a chromosome array, that is, a vector that contains a certain number of control variables. Such variables must satisfy given reciprocal constraints and are used to compute a value of the fitness (or penalty) function. Such a function actually evaluates the scenario elaborated from the "seed" given by the chromosome array following a given ratio. The seed of each iteration is produced according to an evolutionary criterion, and it makes use of both the best chromosomes that have appeared and of some random mutations, which leads to a sudden change in the investigation direction of the algorithm.

For timetabling purposes, once it has been established which and how many trains n_{tr} should be in a given time window, the chromosome array X is composed of $n_{tr} \cdot (n_{st} - 1)$ elements, i.e. the dwell times of each train in all the stations, except the last one.

As previously explained, this seed is sufficient (together with a pre-defined dataset) to write the whole timetable. The latter can be evaluated through a fitness function defined as:

$$f(X) = \sum_{cl=1}^3 (k_{cl} \cdot DW_{cl}) + c1 \cdot \Delta_{cfl}^{c2} \quad (1)$$

$$DW_{cl} = \sum_{tr \in cl} DW_{add,tr} = \sum_{tr \in cl} (DW_{tr} - DW_{min,tr}) \quad (2)$$

$DW_{add,tr}$ rappresenta di fatto il tempo di sosta aggiuntivo globale imposto a ciascun treno lungo il suo viaggio. DW_{cl} è quindi il tempo di sosta aggiuntivo cumulato per ciascuna classe, il quale viene moltiplicato per il fattore di penalità k_{cl} il quale differenzia i contributi delle diverse classi, introducendo così un semplice criterio di priorità.

Δ_{cf} è un termine che va a segnalare l'eventuale presenza di conflitti: esso è nullo se non sono presenti conflitti, mentre diversamente è maggiore di zero. È inoltre proporzionale al numero ed all'entità dei conflitti, dal momento che è calcolato come la somma delle sovrapposizioni degli intervalli di occupazione dei treni coinvolti in un conflitto di orario. I fattori $c1$ e $c2$ sono usati per aumentare drasticamente - alcuni ordini di grandezza - il valore di $f(X)$ ogni qualvolta un orario non sia privo di conflitti: in questo modo tali soluzioni vengono immediatamente scartate dall'algoritmo, il quale mira a minimizzare $f(X)$.

Da un punto di vista concettuale, l'algoritmo genetico è risultato adatto al compito della programmazione: gli orari così generati sono privi di conflitti e *paiono* essere ottimizzati per quanto riguarda i tempi di sosta aggiuntivi. Il verbo "paiono" è qui usato di proposito, dal momento che è stato possibile valutare l'efficacia dell'ottimizzazione solo in casi semplificati, compatibili con la "capacità computazionale" umana. D'altro canto, altri problemi hanno impedito di usare effettivamente l'algoritmo genetico per un'efficace programmazione. Infatti, soprattutto con un elevato numero di treni da gestire - più di 6-8 treni all'ora - l'algoritmo si è rivelato abbastanza lento, dando luogo talvolta a problemi di convergenza. Inoltre, la sua intrinseca natura stocastica fa sì che le soluzioni siano talvolta difficilmente riproducibili, intendendo con questo che in diverse replicazioni il solutore potrebbe fornire - a partire dallo stesso seme iniziale - degli orari realizzabili simili per quanto riguarda il tempo di sosta addizionale complessivo, ma completamente diversi per quanto riguarda lo schema degli incroci. Infine, le priorità tra le classi sono implementate mediocremente, in quanto non è possibile controllare direttamente il processo di risoluzione dei conflitti.

Per queste ragioni, si è preferito abbandonare l'algoritmo genetico preferendo implementare un algoritmo di programmazione che desse la possibilità di controllare il processo più da vicino.

3.2. Un algoritmo euristico greedy

Si è trovata un'alternativa nell'implementazione di un algoritmo di programmazione che di fatto simula le operazioni eseguite da un controllore del traffico (o da un orarista) qualora modifichi manualmente un orario, considerando nello stesso tempo alcuni principi ottimizzativi. Il risultato è un algoritmo *euristico greedy*, il quale risolve i conflitti uno alla volta fino a che l'orario non ne risulta completamente privo. I criteri a tal fine seguiti possono essere facilmente implementati introducendo le priorità tra le classi, permettendo in questo modo un controllo

$DW_{add,tr}$ represents the overall additional dwell time that each train gathers during its journey. DW_{cl} is therefore the cumulated additional dwell time for a whole class, which is multiplied by the penalty factor k_{cl} that differentiates the contributions of the different classes, thus introducing a simple priority ratio.

Δ_{cf} is a term which shows the presence of conflicts: it is null if no conflict exists, otherwise it is greater than zero. Furthermore, it is proportional to the number and the entity of the conflicts, since it is calculated as the sum of the overlaps of the occupation intervals of all the conflictual trains in the timetable. Factors $c1$ and $c2$ are used to drastically raise (by some orders of magnitude) the value of $f(X)$, whenever a timetable is not conflict-free: in this way, such a solution would immediately be discarded by the algorithm, which has the aim of minimizing $f(X)$.

From a conceptual point of view, the genetic algorithm has proved to work for these timetabling purposes: the provided timetables are conflict free and seem to be optimised, with respect to additional dwell times. The verb "seem" has been used intentionally here since it has only been possible to evaluate the optimisation effectiveness in simplified cases, that is, those that are compliant with human "computational power". However, other issues have prevented the genetic algorithm from being adopted for timetabling. The algorithm has in fact resulted to be rather slow, especially when a high number of trains (more than 6-8 per hour) have to be handled, and convergence complications have sometimes emerged. Moreover, its intrinsic stochastic nature sometimes causes the solutions to be difficult to reproduce, which means that, in different replications, the solver, starting from the same initial seed, could provide similar feasible timetables for the global additional dwell time, but completely different ones for the train crossings. Finally, the priorities between classes are poorly implemented, since it is not possible to directly control the conflict-solving process.

For these reasons, it was decided to leave aside the GA and to implement a scheduling algorithm that featured the possibility of controlling the process more closely.

3.2. A heuristic greedy algorithm

An alternative has been to adopt a scheduling algorithm that actually simulates the operations performed by a train dispatcher, while the timetables are arranged manually considering some optimisation principles. The result is a heuristic greedy algorithm, which solves conflicts one after the other until the timetable is completely conflict-free. The ratio adopted for this task can easily be implemented considering class priorities. This allows complete control of the phenomenon and some optimisation functions, which limit the additional dwell time introduced to solve a conflict, to be inserted. Consequently, the proposed algorithm performs a local optimisation in the neighbourhood of the conflict that has to be resolved.

completo del fenomeno ed introducendo nel frattempo alcune funzioni di ottimizzazione, le quali controllano il tempo di sosta addizionale che deve essere introdotto per risolvere un conflitto. Di conseguenza, l'algoritmo proposto esegue un'ottimizzazione locale nell'intorno del conflitto da risolvere.

Il funzionamento dell'algoritmo si basa sullo schema raffigurato in fig. 6. Il punto di partenza è un orario "grezzo", che presenta le tracce da programmare. L'algoritmo lancia il sottoprogramma di ricerca dei conflitti, localizzando quelli presenti. Il primo conflitto della serie (in ordine temporale) è quindi preso in considerazione, analizzato e risolto, rispettando date regole di priorità. A questo punto il sottoprogramma di ricerca è lanciato di nuovo, in quanto la soluzione di un conflitto può generarne di nuovi, e questo processo è iterato fintantoché non ne sono più presenti.

Il criterio portante di questo algoritmo è che *la soluzione di un conflitto non deve generarne di nuovi prima del conflitto in esame*. Questa condizione - unitamente al fatto di risolvere i conflitti in ordine temporale - introduce di fatto una forte euristica nell'algoritmo, ma evita per contro che esso vada in loop. Di conseguenza, le tracce dei treni possono solo essere mosse "avanti" nel tempo, con alcune eccezioni discusse più avanti.

L'algoritmo caratterizza ciascun conflitto secondo il suo tipo:

- i conflitti di tipo 1 sono quelli tra treni marcianti in direzioni opposte su sezioni di calcolo a binario unico;
- i conflitti di tipo 2 sono quelli tra treni che viaggiano nella stessa direzione senza rispettare il minimo distanziamento temporale ammesso nella sezione di calcolo interessata.

La strategia di soluzione cambia significativamente a seconda del tipo di conflitto, ma è in ogni caso basata sul concetto che per risolvere un conflitto uno dei due treni coinvolti deve fermarsi in una stazione rimanendo quindi fermo per un certo tempo di sosta aggiuntivo, il quale rappresenta un ritardo rispetto alla sua marcia indisturbata. La scelta del treno da penalizzare deve quindi essere basata su precise regole di priorità. Per esempio, in caso di un conflitto tra i treni p e q , l'algoritmo calcola entrambi i possibili scenari di risoluzione, vale a dire quello in cui il treno p è penalizzato e quello che invece ritarda il treno q . Per ciascun scenario di soluzione viene calcolato un valore di penalità P_i per il treno ritardato $i \in \{p, q\}$. Tale valore è definito come la combinazione lineare di tre funzioni (criteri di priorità):

$$P_i = c_1 \cdot f_1(cl(i)) + c_2 \cdot f_2(cl(i), D_{add,i}) + c_3 \cdot f_3(cl(i), D_{cum,i}) \quad (3)$$

- f_1 dipende dalla classe $cl(i)$ del treno i ;
- f_2 dipende dalla classe e dal ritardo addizionale $D_{add,i}$ causato al treno i dall'operazione in corso;
- f_3 dipende dalla classe e dal ritardo $D_{cum,i}$ fino a quel

The operation of the algorithm relies on the scheme presented in fig. 6. The starting point is a "raw" timetable that features the train paths that have to be arranged. The algorithm launches the conflict detection subroutine to locate the existing conflicts. The first conflict (in order of time) is then considered, analysed and solved, according to certain prefixed priority rules. At this point, the conflict detection subroutine is run again, since the solution of a conflict could generate new ones, and the process is iterated until no more conflicts are present.

The fundamental criterion which drives this algorithm is that the solution of a conflict should not generate new ones before the conflict in point. This condition - in addition to resolving conflicts in temporal order - actually introduces a strong heuristic aspect into the algorithm, but prevents it from falling into a loop. Consequently, the train paths can only be moved "forward" in time, with the exceptions described hereafter

The algorithm is able to identify each conflict according to its type, namely:

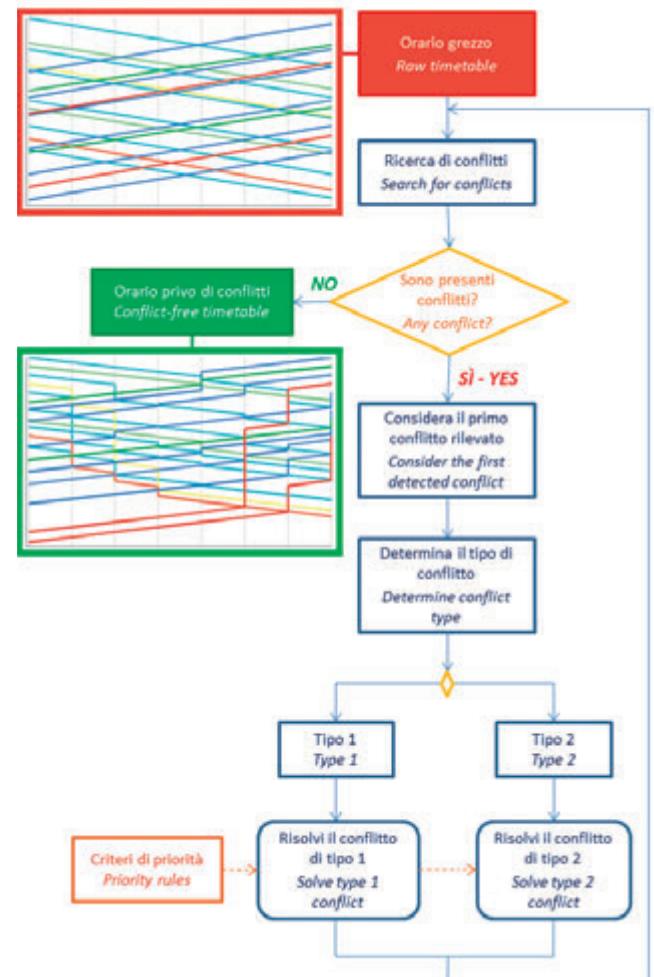


Fig. 6 - Diagramma di flusso dell'algoritmo euristico.

Fig. 6 - Heuristic algorithm workflow.

- punto accumulato dal treno i durante la risoluzione dei conflitti precedenti;
- c_1, c_2 e c_3 sono parametri che possono essere impostati dall'utente pari ad 1 o 0 al fine di attivare o meno i criteri di priorità.

In particolare, f_2 and f_3 sono definite come funzioni continue (diverse per ciascuna classe di treni), il cui andamento può essere monotono o meno, secondo le preferenze dell'utente. Un'opportuna calibrazione di tali funzioni permette di impostare una gran varietà di criteri di priorità. Per esempio, a seconda della definizione di P_i (impostata dall'utente), l'algoritmo potrebbe considerare più vantaggioso fermare per una breve sosta un treno ad alta priorità piuttosto che penalizzarne uno a bassa priorità con una sosta molto lunga - meglio fermare un Intercity per 2 minuti o un treno merci per 20 minuti? O diversamente ancora, l'algoritmo potrebbe scegliere di non penalizzare con ulteriore tempo di sosta un treno che abbia già subito numerose fermate, presentando pertanto un elevato D_{cum} .

Inoltre l'algoritmo esegue una scelta appropriata delle stazioni da impiegare per le fermate. Infatti, esso assume che ciascuna stazione ha un limitata capacità (numero di binari) da assegnare nel tempo. Inoltre si può scegliere se una certa classe può o meno usare una stazione per incrociare altri convogli, per esempio a causa dell'incompatibilità tra la lunghezza dei binari di incrocio e quella dei treni di quella classe. L'algoritmo cerca di risolvere i conflitti fermando i treni nella stazione più vicina, dal momento che in questo modo il ritardo addizionale viene minimizzato. Se questo non è possibile a causa della mancanza di binari ivi disponibili, la procedura è ripetuta con la stazione seguente, e così via.

È opportuno evidenziare un'ulteriore caratteristica di ottimizzazione dell'algoritmo. Si consideri per esempio l'incrocio di due treni p e q (dove p è quello penalizzato). L'algoritmo cercherà di spostare "indietro nel tempo" il treno q il più possibile, al fine di ridurre il ritardo addizionale da attribuire al treno p . In questo modo un conflitto potrebbe addirittura essere risolto semplicemente traslando all'indietro la traccia del treno q . Come stabilito precedentemente, uno dei principi base dell'algoritmo è che nessun nuovo conflitto deve sorgere prima di uno già risolto. Di conseguenza, questa operazione di "traslazione all'indietro" non deve generare nessun nuovo conflitto: questa condizione è soddisfatta traslando contemporaneamente la tracce del convoglio q e di tutti gli altri treni che interferirebbero durante questa operazione, effettuando di fatto la compressione dell'orario antecedente il conflitto in esame. Durante questa operazione, è anche possibile ridurre il tempo di sosta addizionale in stazione precedentemente inserito - e dovuto a conflitti precedentemente risolti - fino a raggiungere un limite inferiore predefinito.

Infine, la compressione all'indietro deve essere interrotta da un limite temporale inferiore, prima del quale le

- type 1 conflicts are between trains running in opposite directions on single-track calculation sections;
- type 2 conflicts are between trains running in the same direction, which do not respect the minimum headway allowed in the calculation section of interest.

The resolution strategy varies significantly, according to the conflict type, but it is always based on the concept that one of the two trains involved has to be stopped in a station and then given a certain amount of additional dwell time in order to solve a conflict. The latter additional time represents a delay with respect to an un-disturbed run. Therefore, the choice of the penalised train should reflect certain prefixed priority rules. In the case of conflict between trains p and q , for instance, the algorithm computes both of the possible solution scenarios, i.e. the one where train p is penalised and the one in which train q is delayed. A penalty value P_i is calculated for each solution scenario, for the delayed train $i \in \{p,q\}$. This value is defined as the linear combination of three functions (priority criteria):

$$P_i = c_1 \cdot f_1(cl(i)) + c_2 \cdot f_2(cl(i), D_{add,i}) + c_3 \cdot f_3(cl(i), D_{cum,i}) \quad (3)$$

- f_1 depends on the class $cl(i)$ of train i ;
- f_2 depends on the class and on the additional delay $D_{add,i}$ caused by the current operation to train i ;
- f_3 depends on the class and on the cumulated delay $D_{cum,i}$ already accumulated by train i during previous conflict resolutions;
- c_1, c_2 and c_3 are parameters which can be set, by the user, equal to 1 or 0 in order to enable or block a priority criteria.

f_2 and f_3 are defined as continuous functions (different for each class of train), whose trend can be monotonic or non-monotonic, according to the user's preference. A proper tuning of such functions permits a large variety of priority ratios to be set up. For instance, according to the current definition of P_i (set by the user), the algorithm could consider it more advantageous to stop a high priority class train for a short time than to penalise a low priority one with a very long dwell time (is it better to stop an Intercity for 2 minutes or a freight train for 20 minutes?). On the other hand, the algorithm could choose not to give more dwell time to a train that has already experienced several stops, and which therefore has a significant D_{cum} .

Furthermore, the algorithm makes an appropriate choice of the stations that have to be used for stops. In fact, it considers that each station has a limited capacity (number of sidings) that can be allocated over a certain time. Moreover, it is possible to establish whether a particular train class can use a station to cross another train, because, for instance, the sidings may be shorter than those trains. The algorithm tries to solve conflicts by stopping trains in the closest available station, since in this way any additional delay is minimised. If this is not possible, be-

tracce dei treni non possono essere più traslate. Senza questo limite ogni conflitto sarebbe risolto semplicemente spostando all'indietro l'intero orario ad esso antecedente. Nelle applicazioni che riguardano la generazione di orari, questo limite inferiore è generalmente fissato all'inizio della finestra temporale considerata. In ogni caso, come verrà spiegato nella sezione 4.3., in altre applicazioni il limite inferiore può essere definito diversamente.

La fig. 7, illustra lo schema delle operazioni eseguite dall'algoritmo per risolvere un conflitto tra due convogli viaggianti in direzioni opposte. Nel caso di un conflitto del secondo tipo, le operazioni sarebbero analoghe: la scelta (scenario 1 o 2) viene ad essere se conservare l'ordine dei due treni consecutivi - ritardando il secondo - o se impostare una precedenza, fermando cioè il primo treno e lasciandolo ripartire una volta superato dal secondo. Inoltre in questo caso l'algoritmo controlla se la stazione più vicina - a monte rispetto al conflitto - ha binari disponibili per tale operazione. In caso contrario, la soluzione del conflitto viene ritentata nella stazione precedente, e così via.

4. Applicazioni dell'algoritmo

4.1. Analisi di capacità basata sugli orari

L'algoritmo di programmazione descritto nella sezione precedente può essere adoperato in varie maniere, di cui la principale è la generazione automatica di orari. Questa funzionalità è molto utile per svolgere valutazioni di capacità basate sugli orari, in alternativa ai metodi analitici. Un'indicazione della massima capacità può essere fornita sotto forma di un orario realizzabile (intendendo con realizzabile – feasible in lingua inglese – l'effettiva capacità di essere applicato nell'esercizio reale, dal momento che non prevede alcun conflitto di circolazione) che contenga un certo numero di tracce orarie o giornaliere con un certo livello di qualità. Un editor di orari automatico, capace di generare in un tempo accettabile (poche decine di secondi) un orario privo di conflitti che rispetti determinati vincoli di priorità costituisce lo strumento ideale per tale compito (fig. 8).

$$\text{orario realizzabile} \leftrightarrow \begin{cases} \text{numero di treni (orario - giornaliero)} \\ \text{mix di traffico (eterotachicità, eteropriorità)} \\ \text{qualità dell'orario (KPI)} \end{cases} \rightarrow \text{stima della capacità}$$

A tal fine si può adoperare l'algoritmo presentato nella sezione 3.2.: l'utente deve fornire come seme di partenza un orario grezzo, nel quale figuri un certo numero di treni n_{cli} per ciascuna classe (*mix di traffico*). Al fine di velocizzare la convergenza dell'algoritmo, è opportuno distribuire omogeneamente i convogli nella finestra temporale considerata, evitando di concentrarli solo in certi intervalli.

Se si imposta il problema in questo modo, la soluzione dipende dal mix di traffico oltre che dal seme dei tem-

cause no siding is available, the procedure is repeated for the subsequent station.

It is worth highlighting another optimisation feature of the algorithm. If, for instance, the crossing of two trains p and q (where p is the penalised one) is considered, the algorithm tries to move train q as far "backward in time" as possible, in order to reduce the additional delay given to train. This way, it might also be possible to solve the conflict by just moving train backwards. As previously mentioned, one of the algorithm's principles is that no new conflicts should arise before another one has been solved. Consequently, this "moving upstream" operation should not generate any new conflict: this is eventually obtained by shifting the paths of train q, and of all the other trains that would interfere in this operation, together, thus actually performing a compression of the timetable before the point of conflict. During this operation, it is also possible to reduce the additional, previously introduced dwell time in the station (because of already solved conflicts) to a pre-defined minimum value.

Finally, the backward compression should be stopped at a temporal lower boundary, before which it is not possible to move the train paths any further. Without this boundary, each conflict could be solved simply by shifting the whole previous timetable backwards. In timetable-generation applications, this lower boundary is generally set at the beginning of the period that has to be covered. However, as explained in section 4.3., the lower boundary could be set differently for other applications.

Fig. 7 shows a workflow of the operations performed by the algorithm to solve a conflict situation between two trains running in opposite directions. In the case of conflict of the second type, the operations would be similar: the choice (scenarios 1 or 2) would be whether to maintain the order of the two consecutive trains (delaying the second one) or to set up an overtaking operation, i.e. stopping the first train and letting it depart after the second one had passed. In addition, in this case, the algorithm checks whether the closest station (upstream to the conflict) has the necessary capacity for this operation. If not, the solution is tried for the previous station, and so on.

4. Algorithm applications

4.1. Timetable-based capacity assessment

The scheduling algorithm described in the previous section can be used for several applications. The main one pertains to the automatic generation of train timetables, which can be a very useful function for carrying out timetable-based capacity evaluations, as an alternative to analytical methodologies. An indication of the maximum capacity can be provided by a feasible timetable – where feasible means that it could actually be set up, since no conflict is foreseen featuring a certain number of hourly or daily trains and a certain quality level. An automatic timetable editor, suitable for generating a conflict-free timetable in an

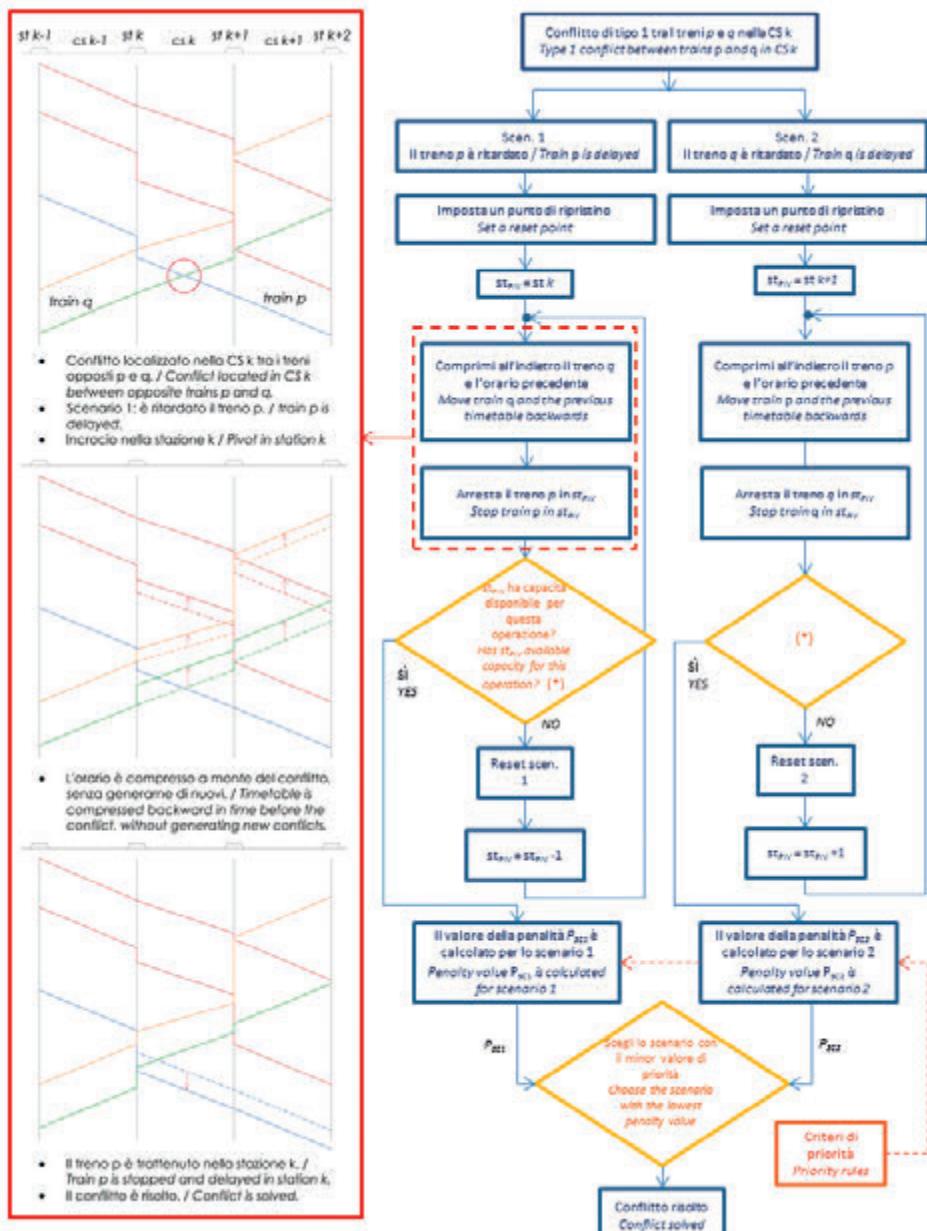


Fig. 7 - Diagramma di flusso per la soluzione di un conflitto di tipo 1 (treni viaggianti in direzioni opposte).
 Fig. 7 - Resolution workflow of a type 1 conflict (trains running in opposite directions).

pi di entrata. Bisogna sottolineare che l'algoritmo presentato raggiunge sempre una soluzione, fornendo un orario realizzabile a prescindere dal numero complessivo dei convogli. Nel caso che il loro numero superi la capacità effettiva della linea, essi verrebbero semplicemente "parceggiati" nelle stazioni intermedie in attesa di uno slot di tempo per raggiungere la stazione successiva. Se le stazioni intermedie non hanno sufficiente capacità disponibile (per esempio nel caso in cui i loro binari siano già occupati), i convogli vengono trattenuti nelle stazioni di estremità (le quali sono modellizzati con un'abilità virtuale illimitata al fine di assicurare la convergenza dell'al-

acceptable time (a few tens of seconds) while respecting certain previously established priority constraints, represents the key tool for this task (fig. 8).

$$\text{orario realizzabile} \leftrightarrow \begin{cases} \text{number of trains (trains per our - day)} \\ \text{traffic mix (speed/priority heterogeneity)} \\ \text{timetable quality (KPIs)} \end{cases} \rightarrow \text{capacity evaluation}$$

The algorithm presented in section 3.2 can be used for this purpose: the user should provide a "raw" timetable, featuring a certain number of trains n_{cli} for each class (traffic

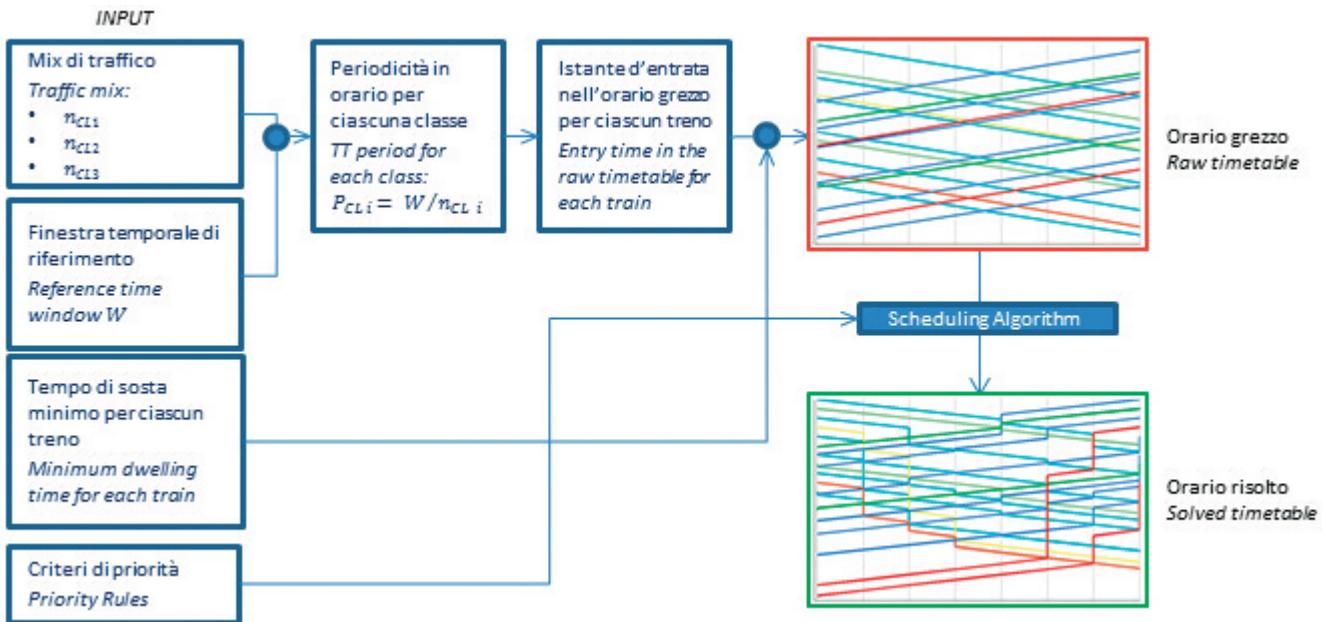


Fig. 8 - Diagramma di flusso per generazione automatica di orari.

Fig. 8 - Automatic timetable generation workflow.

goritmo), posticipando quindi l'uscita dell'ultimo treno dalla linea.

Gli orari realizzabili sono valutati con una serie di *Key Performance Indexes*, utili per stimare come e quanto sia utilizzata la capacità disponibile. Una lista di KPI adatti allo scopo, già presentata e discussa approfonditamente in [15] e [16], può essere richiamata ed estesa come segue:

1. numero delle tracce programmate;
2. mix di traffico;
3. entità degli allungamenti di percorrenza;
4. entità dei margini di regolarità;
5. occupazione dell'infrastruttura secondo la fiche UIC 406;
6. tempo di viaggio programmato di ciascun treno, o tempo di sosta globale programmato per ciascun treno;
7. resilienza dell'orario.

Le prime tre voci sono di fatto i dati di input dell'algoritmo di programmazione. Il numero di treni in orario e la composizione del mix di traffico sono caratteristiche esplicite del "seme" d'orario, mentre gli allungamenti di percorrenza possono essere implicitamente contenuti nel parametro rt del dataset ovvero introdotti esplicitamente da K_{rt} (si veda la sezione 2.3). Il tempo cuscinetto desiderato viene introdotto dall'algoritmo tra ciascun paio di tracce consecutive - rivolte in direzione eguale od opposta - ogniqualvolta viene risolto un conflitto.

L'occupazione dell'infrastruttura dipende principalmente dal mix di traffico e dal numero di treni complessivo, ma anche dai loro ordine e alternanza (la quale può

mix), as a starting seed. In order to speed up the convergence of the algorithm, it is opportune to distribute the trains evenly in the considered time window, in other words, to avoid concentrating them in only certain time slots.

If the problem is set up this way, the solution depends on the traffic mix, but also on the entry time seed. It is worth pointing out that the presented algorithm always reaches a solution, and arranges a feasible timetable, whatever the overall train number is. In the case of a higher number of trains than that of the capacity of the actual line, the extra trains would simply be parked in an intermediate station to wait for a free time slot before travelling to the next available one. If the intermediate stations do not have sufficient available capacity (i.e. their sidings are already occupied), the trains are kept in the starting and ending stations (which are modelled with a virtual unlimited capacity in order to ensure convergence of the algorithm), thus postponing the exit of the last train from the timetable.

Feasible timetables are evaluated through a series of *Key Performance Indicators*, which are able to estimate how and to what extent the available capacity is utilised. A list of suitable KPIs is presented and discussed in detail in [15] and [16], and it is recalled and extended as follows:

1. number of scheduled trains;
2. traffic mix;
3. number of run time supplements;
4. amount of scheduled buffer time;
5. infrastructure occupation according to UIC code 406;
6. scheduled travelling time for each train, or scheduled total dwell time for each train;

variare nell'orario a causa del processo di riprogrammazione). L'occupazione dell'infrastruttura si può calcolare, in conformità con la fiche UIC 406 [17], per mezzo della formula proposta [5], capitolo 2. In ciascuna sezione di calcolo k , il coefficiente di occupazione η_k è definito come

$$\eta_k = \frac{n \cdot \sum(h_{k,ij} \cdot f_{k,ij})}{t_p} = \frac{n \cdot \sum(h_{k,ij} \cdot \frac{n_i \cdot n_j}{n^2})}{t_p} \quad (4)$$

dove:

- n è il numero di treni totale;
- $h_{k,ij}$ è il distanziamento minimo ammesso nella CS k tra un treno di classe i ed uno di classe j (in tale ordine). Tale parametro è contenuto nel dataset mesoscopico;
- $f_{k,ij}$ è la frequenza relativa della combinazione “treno j segue treno i ”;
- n_i e n_j sono il numero dei treni rispettivamente delle classi i e j ;
- t_p è la finestra temporale presa in considerazione.

Questa formula determina $f_{k,ij}$ in modo statistico. Inoltre non considera esplicitamente la circolazione bidirezionale sulle linee a singolo binario. Infatti tiene conto solo del distanziamento h_A tra treni viaggianti nella stessa direzione. Seguendo l'approccio analitico descritto in [2], in caso di circolazione bidirezionale bisogna definire anche un distanziamento h_B per le coppie di treni i e j che viaggiano in direzioni opposte. Considerando che il treno j transita dopo il treno i , questo distanziamento può essere calcolato per mezzo dei parametri del dataset come

$$h_{k,B,ij} = t_{o2,k,i} + t_{o1,k,j} \quad (5)$$

Avendo a disposizione l'orario, le frequenze $f_{k,ij}$ possono essere ricavate direttamente contando l'alternanza delle classi e delle direzioni dei treni. Si ottiene pertanto la formula

$$\eta_k = \frac{n \cdot [\sum(h_{k,A,ij} \cdot f_{k,A,ij}) + \sum(h_{k,B,ij} \cdot f_{k,B,ij})]}{t_p} \quad (6)$$

dove:

- $f_{k,A,ij}$ è la frequenza relativa della combinazione “il treno j segue il treno i viaggiando nella stessa direzione”;
- $f_{k,B,ij}$ è la frequenza relativa della combinazione “il treno j e il treno i viaggiano in direzioni opposte, ed il treno j entra nella CS dopo il treno i ”.

I KPI numero 6 e 7 possono essere visti come i veri output del processo di generazione di orari. Il tempo di viaggio programmato è di fatto il tempo richiesto da ciascun treno per attraversare la linea studiata. È composto dal tempo di percorrenza effettivo, dai minimi tempi di so-

7. timetable resilience.

The first four items are actually used as input data for the scheduling algorithm. The number of scheduled trains and the composition of the traffic mix are explicit characteristics of the timetable “seed”, while run time supplements can be implicitly contained in the dataset parameter rt or explicitly introduced by K_{rt} (see section 0). The desired scheduled buffer time is introduced by the algorithm between each pair of consecutive trains (running in both the same or in opposite directions) whenever it solves a conflict situation.

Infrastructure occupation mainly depends on the traffic mix and on the overall number of trains, but also on the train order and alternation (which can vary over the timetable as a result of the rescheduling process). Infrastructure occupation can be calculated, according to UIC code 406 [17], through the formula proposed in [5], chapter 2. The occupation ratio η_k in each calculation section k is defined as

$$\eta_k = \frac{n \cdot \sum(h_{k,ij} \cdot f_{k,ij})}{t_p} = \frac{n \cdot \sum(h_{k,ij} \cdot \frac{n_i \cdot n_j}{n^2})}{t_p} \quad (4)$$

where:

- n is the total number of trains;
- $h_{k,ij}$ is the minimum headway in CS k between a train in class i followed by one in class j . This parameter is available in the mesoscopic dataset;
- $f_{k,ij}$ is the relative frequency of the “train j following train i ” combination;
- n_i and n_j are the number of trains in class i and j , respectively;
- t_p is the considered time window.

This formula makes use of a statistical process to determine the term $f_{k,ij}$. Furthermore, it does not take into explicit consideration bidirectional operation on single-track lines. In fact, it just considers the headways, h_A between trains running in the same direction. According to the analytical approach described in [2], for bidirectional operation, it is necessary to introduce headway h_B between a couple of trains, i and j which run in opposite directions. If train j follows train i , this headway can be calculated by resorting to the following dataset parameters:

$$h_{k,B,ij} = t_{o2,k,i} + t_{o1,k,j} \quad (5)$$

Once the timetable is available, the frequencies $f_{k,ij}$ can be obtained directly by counting the train alternations pertaining to the class and the running direction. The resulting formula is therefore

$$\eta_k = \frac{n \cdot [\sum(h_{k,A,ij} \cdot f_{k,A,ij}) + \sum(h_{k,B,ij} \cdot f_{k,B,ij})]}{t_p} \quad (6)$$

sta - richiesti, per esempio, dal servizio passeggeri - e dai tempi di sosta aggiuntivi. Quest'ultimi sono i periodi in cui il treno deve rimanere fermo in stazione a causa di incroci o precedenze. Poiché il tempo di percorrenza è un parametro costante definito dal dataset ed il tempo di sosta minimo è fissato a priori (almeno in prima approssimazione e trascurando le perturbazioni - si veda la sezione 4.3.), risulta indifferente valutare il tempo di viaggio totale piuttosto che il tempo di sosta aggiuntivo complessivo.

La fig. 9 illustra l'andamento del tempo di sosta aggiuntivo nelle stazioni intermedie (valor medio per i treni di ciascuna classe) per un insieme di orari ricavati inserendo un numero crescente di treni in una finestra temporale di 4 ore. Il grafico evidenzia come la massima priorità (implicante il minimo tempo di sosta aggiuntivo DW_{add}) sia riservata per i treni di classe 2, seguiti da quelli di classe 1. D'altro canto, la classe 3 ha la minima priorità, significativamente inferiore alle altre. Fino a circa 20 treni/4 h il tempo di sosta aggiuntivo aumenta linearmente, mentre l'istante di uscita dell'ultimo treno in orario viene posticipato in maniera meno che proporzionale. Questo significa che l'orario sta venendo saturato "internamente", aumentando la densità delle tracce insieme ai tempi di sosta nelle stazioni intermedie. Diversamente, sopra il limite dei 20÷25 treni/4 h il grafico denuncia la saturazione dell'orario, dal momento che DW_{add} rimane piuttosto costante e l'istante di uscita dell'ultimo treno subisce una repentina posticipazione. Questo accade perché ciascuna traccia aggiuntiva viene posticipata, dal momento che non può più trovare collocazione nelle slot di tempo precedenti.

Il limite di saturazione marcato in fig. 9 con un linea tratteggiata rossa è quello massimo teorico, il quale può essere visto come la massima capacità teorica ottenibile con date regole di priorità. Ad ogni modo, questo limite non tiene conto dei vincoli legati alla qualità del servizio, i quali verosimilmente potrebbero richiedere di contenere il tempo di sosta aggiuntivo o di favorire la stabilità dell'orario con tempi cuscinetto adeguati tra le tracce. Un limite di saturazione a qualità può essere quindi definito come il più alto numero di treni (nella finestra temporale considerata) che soddisfi le seguenti condizioni:

- il tempo di sosta aggiuntivo globale - ovvero il tempo di sosta aggiuntivo per chilometro percorso - deve essere minore di un certo valore;
- il coefficiente di occupazione dell'infrastruttura deve essere minore dei limiti raccomandati dalla fiche UIC 406 [17];
- la stabilità dell'orario in determinate condizioni di perturbazione deve essere assicurata (si veda la sezione 4.3.).

L'analisi di un singolo orario realizzabile potrebbe essere fuorviante, in quanto la capacità stimata in questo modo dipenderebbe anche dai vincoli interni introdotti dall'orario in sé, diversamente da un metodo indipendente da orari. Una possibile soluzione per superare questa im-

where:

- $f_{k,A_{ij}}$ is the relative frequency of the "train j following train i running in the same direction" combination;
- $f_{k,B_{ij}}$ is the relative frequency of the "train j and train i running in opposite directions, and train j entering CS k after train i " combination.

KPIs n° 6 and 7 can be considered as the real outputs of the scheduling process. The scheduled travel time is actually the time needed by each train to cross the calculation line. It is composed of the run time, the minimum dwell time (necessary, for example, for the passenger service) and the additional dwell time. The latter is the time that trains must spend idle in stations due to crossings or overtaking. As the run time is a constant parameter of the dataset and the minimum dwell time is a fixed value (at least in the first instance, neglecting any perturbation – see section 4.3.), it is just the same to evaluate the total travel time or the total additional dwell time.

Fig. 9 shows the additional dwell time in intermediate stations (average values per train for each class) for a set of timetables obtained with a rising number of trains entering a 4-hour time window. The graph highlights how the maximum priority (lowest additional dwell time DW_{add}) is given to class 2 trains, and this is immediately followed by class 1 ones. On the other hand, class 3 trains have the lowest priority, which is significantly lower than the others. The additional dwell times rise rather linearly, up to approximately 20 trains/4h, while the exit time of the last train in the timetable increases less than proportionally. This means that the timetable is being saturated "internally", that is, the density of the train paths is increasing as well as the stop times in intermediate stations. The graph shows saturation of the timetable above the limit of 20÷25 trains/4h, since DW_{add} remains rather constant and the exit time of the last train displays a sharp rise. This happens because any additional train is postponed, since no more space is available in the preceding time slots.

The saturation limit marked with a red dashed line in fig. 9 is the theoretical maximum one, which can be considered as the maximum theoretical capacity with certain priority rules. However, this limit does not consider service quality issues, which would likely require the additional dwell time to be contained or the timetable stability to be enhanced with a sufficient buffer time between the train paths. The quality saturation limit can therefore be determined as the highest number of trains (in the considered time window) which fulfils the following conditions:

- the global additional dwell time (or the additional dwell time per travelled kilometre) is less than a given value;
- the track occupation ratio is lower than the recommended limits presented in UIC code 406 [17];
- timetable stability, under a given perturbation condition, is ensured (see section 4.3.).

The analysis of a single feasible timetable could be misleading, because the thus-estimated capacity would also de-

passee consiste nell'analizzare un appropriato insieme di orari realizzabili, al fine di mediare le possibili oscillazioni degli indicatori dovuti da situazioni contingenti. In questo modo i risultati globali sono ottenuti estrapolando opportuni *indicatori statistici* dall'insieme delle replicazioni.

Tale metodo richiede di calcolare un gran numero di orari realizzabili e privi di conflitti. Tale numero può essere ulteriormente aumentato dall'eventuale necessità di considerare più di un mix di traffico, facendo sì che questo compito possa essere eseguito solo tramite un procedura di timetabling automatico.

Questo approccio statistico viene applicato perturbando un "seme" di base (un orario grezzo ancora da risolvere), spostando gli istanti di entrata dei vari treni di incrementi positivi o negativi estratti casualmente da un dato intervallo. Risolvendo poi questi orari grezzi, si ricava un pari numero di orari realizzabili i quali presentano il medesimo mix di traffico ma un diverso arrangemento delle tracce. In questo modo si può ricavare un insieme di valori per ciascun parametro, dal quale possibile estrarre degli indicatori statistici come il valor medio, la deviazione standard o i valori massimi e minimi.

In [16] la resilienza è definita come "*la flessibilità di un orario nel prevenire o ridurre i ritardi secondari attraverso la riprogrammazione attiva delle tracce (dispatching)*". Questa definizione riguarda esplicitamente la gestione della circolazione perturbata. In ogni caso, questo concetto può essere esteso anche alla fase di programmazione e progetto dell'orario, che può essere vista come l'operazione di *dispatching* su un seme d'orario grezzo. Da questo punto di vista, la resilienza di una certa infrastruttura con date regole di priorità si può definire come la possibilità di progettare diversi orari (schemi di incroci e precedenze) senza assistere nel contempo a variazioni significative nel tempo di sosta aggiuntivo. Tale caratteristica permette una più flessibile programmazione dell'esercizio, facilmente gestibile in caso di perturbazioni attraverso un attivo controllo della circolazione (riconlegandosi quindi alla definizione data in [16]).

Un insieme di orari realizzabili (ottenuti tramite la procedura di perturbazione di un seme grezzo) che non presenti oscillazioni significative dei valori di DW_{add} (bassa deviazione standard), rivela una condizione di resilienza di quel sistema ferroviario (composto da infrastruttura e regole d'esercizio).

La fig. 10 riporta un esempio di tale approccio. Tale figura illustra i risultati della valutazione degli effetti di due modifiche infrastrutturali su una linea campione a binario singolo, vale a dire il potenziamento del sistema di segnalamento (riduzione dell'distanziamento minimo,

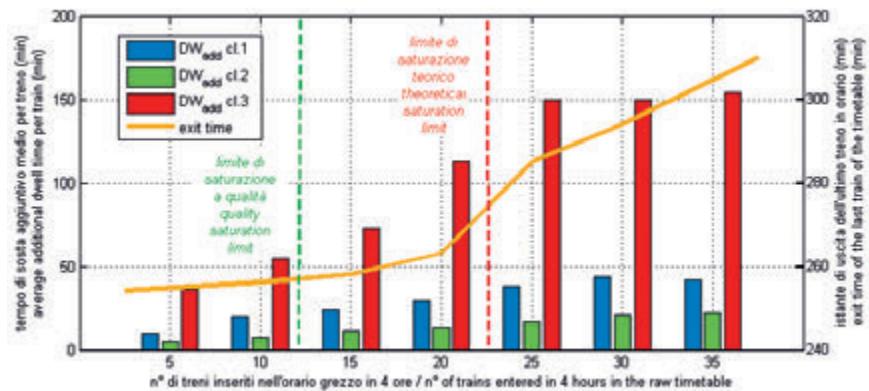


Fig. 9 - Tempo di sosta aggiuntivo e saturazione dell'orario.

Fig. 9 - Additional dwell time and timetable saturation.

pend on the internal constraints introduced by the timetable itself, unlike the case of a timetable-independent methodology. A possible way of overcoming this impasse could be to analyse not just a single timetable, but a proper set of feasible ones, in order to average any possible oscillation of the indicators due to individual situations. Therefore, the overall capacity result would be obtained by extracting the proper statistical indicators from the replication set.

Such a method requires the calculation of a large number of feasible, conflict-free timetables. This number is further increased by the possible necessity of considering more than one traffic mix. It is therefore evident that such a task can only be performed by an automatic timetabling procedure.

This statistical approach is performed by perturbing a basic seed (a raw timetable to be solved), and then moving the entry times of the various trains by positive or negative increments randomly extracted from a given interval. By solving these raw timetables, a number of feasible ones, which feature the same traffic mix, but differently arranged patterns, are obtained. This way, a set of values (instead of a single one) is provided for each indicator, and from these it is possible to extract statistical indicators, such as the mean value, the standard deviation or the maximum/minimum values.

In [16], timetable resilience is defined as "the flexibility of a timetable to prevent or reduce secondary delays using dispatching (re-timing, re-ordering, re-routing)". This definition refers explicitly to the management of perturbed traffic. However, this concept can also be extended to the timetable design stage, since it can be seen as the dispatching of the "raw" timetable seed. From this point of view, the resilience of a certain railway line, with prearranged priority rules, pertains to the possibility of drawing up different timetables (crossing patterns) without any significant modifications of the additional dwell times. Such a configuration would lead to a very flexible operation planning and could easily be re-arranged in the case of perturbations through active dispatching (thus conforming with the definition given in [16]).

SS1, SS2, SS3) e l'incremento del numero di stazioni (posti d'incrocio). Si considerano tre mix di traffico, con un numero di treni complessivo crescente. I risultati vengono forniti in termini di percentuale di tempo di sosta aggiuntivo (valor medio su tutti i treni dell'orario) calcolata rispetto alla situazione peggiore, la quale si manifesta nello scenario di base, con SS1 e il mix di traffico numero 3 - quello maggiormente congestionato. Ciascuno dei $3 \cdot 3 \cdot 2 = 18$ scenari risultanti è stato valutato tramite 25 diverse replicazioni, il cui valore medio e minimo sono riportati nel grafico insieme con gli intervalli di variazione forniti dalla loro deviazione standard.

Il grafico rivela che anche nella "peggiore" situazione infrastrutturale (scenario di base, SS1) non viene raggiunta la saturazione, in quanto i tempi di sosta medi dei vari mix di traffico appaiono ancora differenti. Pertanto in questi casi i benefici dei miglioramenti infrastrutturali riguarderanno principalmente la qualità del servizio. Ad ogni modo è opportuno puntualizzare come il maggior numero di stazioni intermedie dello scenario 2 vada ad innalzare decisamente il limite di saturazione, come sarebbe possibile rilevare analizzando eventuali mix di traffico ancora più congestionati. È altresì possibile apprezzare l'incremento di resilienza generato principalmente dall'aumento del numero di stazioni: in questi scenari i risultati sono maggiormente concentrati attorno al valor medio e gli intervalli di variazione sono di minore entità.

4.2. Analisi della saturazione di un orario

Le applicazioni fin qui descritte riguardano una modalità di valutazione della capacità che potrebbe essere definita come "pura", dal momento che si basa su una generazione di orari in assenza di vincoli importanti. Diver-

Therefore, a set of feasible timetables (obtained using the seed-perturbation procedure) that does not feature significant oscillations in the DW_{add} values (low standard deviation) should result in a resilient condition of that railway system (composed of the infrastructure and operating rules).

An example of this approach is reported in fig. 10, which shows the results of an evaluation of the effects of two infrastructural modifications on a sample single-track line, i.e. an improvement in the signalling system (headway reduction, SS1, SS2, and SS3) and an increase in the number of stations (crossing locations). Three traffic mixes, considering an increasing number of entered trains, have been considered. The results are given in terms of the percentage of the additional dwell times (average value of all the trains in the timetable) with respect to the worst situation, which occurs in the baseline scenario, for SS1 and traffic mix 3 (the most congested one). Each of the resulting $3 \cdot 3 \cdot 2 = 18$ scenarios has been analysed using 25 different replications, whose minimum and mean values are reported in the plot, together with the variation intervals given by their standard deviation.

The graph highlights that saturation is not reached, even in the "worst" infrastructural situation (baseline scenario, SS1), since the average dwell times of the various traffic mixes are still significantly different. Therefore, in this case, the benefits of the infrastructural improvements of interest mainly pertain to service quality, as already explained. However, it is worth pointing out that the greater number of intermediate stations in scenario 2 would significantly increase the saturation limit, as it would emerge with more congested traffic mixes. It is also possible to appreciate the gain in timetable resilience, which is mainly provided by the increase in the number of stations: in these scenarios, the results are more concentrated around the mean values, and the variation intervals are smaller.

4.2. Timetable saturation analysis

The so far described applications refer to a kind of capacity assessment that could be defined as "pure", since it relies on a rather unconstrained timetable generation. However, it could be useful to provide an evaluation that considers explicit timetable constraints, for example, those that respect given train paths, which should not be touched by the algorithm.

For this purpose, the simulation environment allows a train path to be defined as fixed, thus preventing the scheduling algorithm from performing any arrangement action. This function makes it possible, for instance, to check how a given set of additional trains can be superimposed onto a fixed, pre-edited timetable. Again in

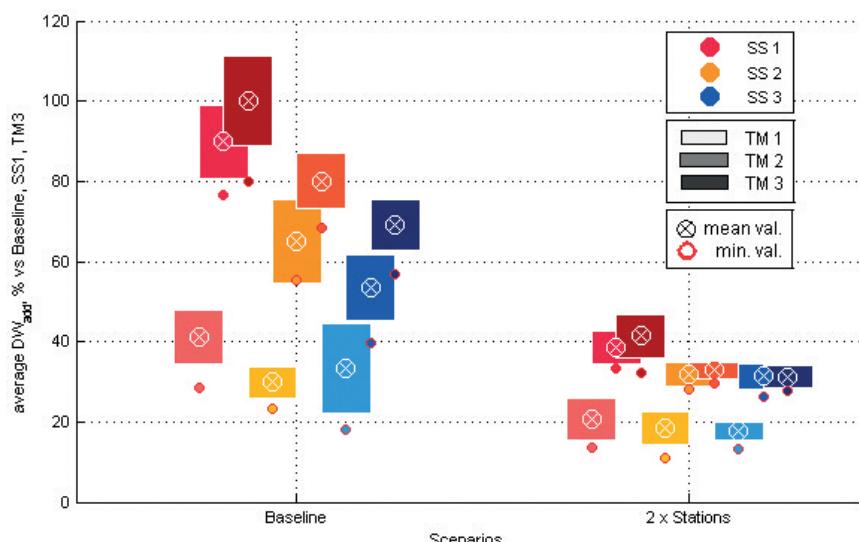


Fig. 10 - Esempio dell'output di un'analisi statistica.
Fig. 10 - Example of statistical analysis output.

samente, può spesso essere utile effettuare una valutazione che consideri esplicitamente dati vincoli d'orario, rispettando per esempio delle tracce che devono essere considerate come "intoccabili" dall'algoritmo.

A tal fine, l'ambiente di simulazione permette di definire una traccia come *fissa*, preservandola così da eventuali modifiche ad opera dell'algoritmo di programmazione. Tale funzionalità permette, per esempio, di verificare se un certo insieme di tracce aggiuntive possa essere sovrapposto ad un orario fissato. Anche in questo caso, si può effettuare un'analisi statistica, usando vari insiemi di tracce generate casualmente nell'intorno di un insieme di base.

Un'applicazione peculiare è rappresentata dall'analisi a saturazione di un dato orario con un certo tipo di treno. Questo tipo di valutazione è utile per determinare, per esempio, le finestre temporali più consone all'inserimento di tracce aggiuntive. Una procedura automatizzata implementa questa funzionalità, con la possibilità di impostare diverse proporzioni di convogli da inserire nelle due direzioni. Le tracce aggiuntive risultanti possono essere di conseguenza classificate attraverso due parametri relativi ad una certa griglia temporale (per esempio, ad intervalli di un'ora):

- la frequenza delle partenze dei treni aggiuntivi dalla prima stazione presente in orario;
- il tempo di sosta aggiuntivo di ciascuna traccia addizionale (valor medio per i treni partiti in un certo slot di tempo).

Questi parametri esprimono l'effettiva possibilità di inserire tracce aggiuntive in un orario esistente con un certo livello di qualità del servizio. La fig. 11 illustra graficamente questo approccio:

- tra le 12:00 e le 14:00 non è possibile inserire tracce aggiuntive;
- tra le 14:00 e le 18:00 è possibile inserire 1 o 2 treni all'ora, i quali d'altro canto devono farsi carico di un significativo tempo di sosta aggiuntivo. Ciò rivela che in quella zona l'orario è già piuttosto saturo.

Questo tipo di approccio può essere uno strumento utile ed efficace per condurre rapide indagini preliminari riguardo a soluzioni di saturazione. Tali risultati possono essere raffinati iterativamente, cancellando per esempio le tracce che presentino un insufficiente livello di qualità e riprogrammando poi le rimanenti. In questo modo si riducono le interazioni reciproche, permettendo di arrangiare più liberamente le tracce preservate.

4.3. Analisi di perturbazione

Fin qui è stata presentata un'implementazione dell'algoritmo di programmazione sviluppata per un'applicazione di progettazione di orari. In tale contesto, dapprima viene generato un orario grezzo che viene poi risolto una volta per tutte dall'algoritmo, il quale risolve tutti i

this case, it is also possible to perform a statistical analysis, using a different set of additional trains randomly generated in the neighbourhood of a base set.

The saturation analysis of a given timetable with a certain train type represents a particular application. This kind of evaluation is useful for investigating, for instance, the most suitable time slots for extra-train paths. An automatic procedure provides this function and offers the possibility of setting different proportions of trains in the two directions. The resulting extra paths can subsequently be classified through two parameters relevant to a certain time grid (for instance, one-hour intervals) throughout the day:

- *the frequency of departures of additional trains from the first station in the timetable;*
- *the additional dwell time of each additional train (the average value of the trains that have departed in the considered time slots).*

These parameters can represent the actual possibility of inserting extra trains in an already existing timetable with a certain service quality. Fig. 11 reports a graphical example of such an approach:

- *between 12:00 and 14:00, there is no space for an extra train;*
- *between 14:00 and 18:00, it is possible to insert 1 or 2 trains/hour, which should be charged with significant additional dwell time. This is a sign that the timetable is already rather saturated.*

This kind of approach could be a useful and effective tool for carrying out quick, preliminary investigations on saturation solutions. The results could then be refined iteratively, for example by deleting the paths that feature poor service quality and by rescheduling the other ones. In this way, mutual train interactions could be reduced, and a better arrangement of the preserved paths could be obtained.

4.3. Perturbation analysis

The scheduling algorithm has so far been presented for a timetable-editing application. In such a context, first a raw timetable is generated and then the algorithm is run once in order to solve any conflicts that are present. This task can be considered as a simulation of the job of the timetable planner, who is in charge of designing a feasible timetable for a certain line.

On the other hand, the same algorithm can be adopted within an environment that simulates the operations of a dispatching centre, i.e. of the logics that manage the actual train traffic. In this way, it is possible to assess the effect of the algorithm itself (which, in this environment, becomes a re-scheduling one) when the traffic is perturbed by various delays.

For this purpose, the scheduling algorithm has been integrated in a discrete event simulation process (fig. 12). This process is based on a temporal succession of events composed of a perturbation action (e.g. a dwell time exten-

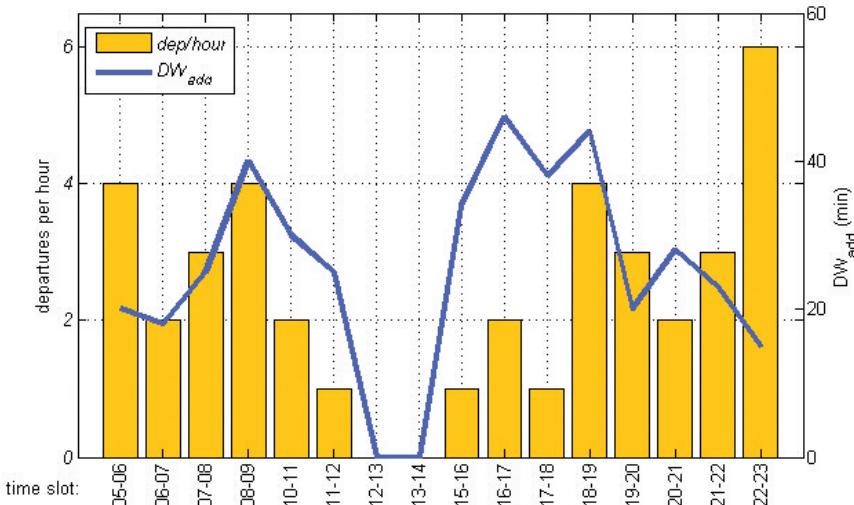


Fig. 11 - Esempio di diagramma di saturazione.

Fig. 11 - Example of a saturation diagram.

conflitti presenti. Questa procedura può essere vista come la simulazione del lavoro di un orarista, incaricato di progettare un orario realizzabile per una data linea.

D'altro canto, lo stesso algoritmo può essere impiegato in un ambiente che simuli l'azione di un *centro di controllo della circolazione*, vale a dire delle logiche che gestiscono effettivamente il traffico ferroviario. In questo modo è possibile valutare l'efficacia dell'algoritmo in sé - che in questo ambiente diventa di ri-programmazione - qualora la circolazione sia perturbata da vari ritardi.

Per questo scopo l'algoritmo di programmazione viene integrato in un processo di simulazione ad eventi discreti (fig. 12), basato sulla successione temporale di eventi composti da un'azione di perturbazione (per esempio il prolungamento del tempo di sosta di un certo treno in una certa stazione) seguita da una di riprogrammazione, la quale simula la reazione del sistema. Si possono applicare vari tipi di perturbazione, come ritardi in entrata o prolungamenti del tempo di percorrenza e di sosta, mentre la loro estensione è estratta casualmente da determinate distribuzioni statistiche (esponenziale negativa, gaussiana, lognormale, ...). È inoltre possibile definire per ciascuna classe e direzione di marcia la percentuale di treni da ritardare - estratti casualmente dall'insieme complessivo - come anche specificare individualmente i treni affetti da perturbazione.

Il tipo di perturbazione, la sua durata e posizione (nello spazio) sono preliminarmente determinati per ciascun treno da ritardare. Su questa base, viene compilata una lista delle perturbazioni - che rappresentano di fatto gli eventi discreti - ordinata nel tempo. Dopo ciascun evento - perturbazione e riprogrammazione - la lista viene aggiornata, in quanto l'orario potrebbe essere stato significativamente cambiato, alterando così la sequenza temporale degli eventi.

sion in a certain station for a certain train) followed by a rescheduling, which simulates the reaction of the system. Various types of perturbation are available, such as entry delays, run times and dwell time prolongation, while their extent has been randomly extracted from a given statistical distribution (negative exponential, Gaussian, lognormal ...). It is also possible to define, for each class and direction, the percentage of trains that should be delayed (randomly extracted from the whole set), as well as to individually specify each train that has to be perturbed.

The perturbation type, extension and location (in space) are preliminarily determined for each train affected by delays. Using this approach, a perturbation list (which is actually the list of the discrete events) is drawn up in temporal order. After each event (perturbation and rescheduling), the list is updated, since the timetable could have changed, thus altering the temporal sequence of the events.

The timetable resulting from this process is therefore the outcome of a series of successive modifications, which are, on one hand, determined randomly on the basis of a given fixed statistical distribution. For this reason, an appropriate analysis has to be carried out considering a set of replications, characterised by the same perturbation type, and then averaging the results, in a similar way to what was proposed in section 4.1. Partial results are evaluated at the end of each replication; they can be represented by various parameters that describe the way in which the timetable has reacted to the applied perturbation. Since the tool can easily compare the perturbed train paths with the scheduled ones, it is possible to obtain both a primary (i.e. that directly applied by the discrete-event simulation) and a secondary delay (the one resulting from the interaction with other delayed trains) for each train. Having this information available, disparate further analysis can be developed, extracting several indicators that have been described extensively in literature ([5], [15], [16]).

Within this framework, it is possible not only to evaluate the reaction of a timetable itself to various perturbation conditions, but also to assess the influence of the adopted rescheduling ratio. The latter can be used to model the signalling and control system of the considered line. It is worth pointing out that such a way of modelling, as developed in the presented mesoscopic environment (which, for example, does not explicitly consider the block sections but only their effect on headways), cannot substitute microscopic simulations that must be used to replicate the operations of each component of the railway system. Therefore, the presented tool can simulate the operations of a dispatching centre with a certain approximation, which is

L'orario che risulta da questo processo è dunque il prodotto di una serie di modifiche successive, le quali sono d'altro canto generate in modo casuale basandosi su una certa distribuzione statistica. Per questi motivi è necessario svolgere l'analisi considerando un insieme di replicazioni caratterizzate dallo stesso tipo di perturbazioni. I risultati vengono poi mediati, secondo una procedura analoga a quanto proposto nella sezione 4.1. Alla fine di ciascuna replicazione vengono valutati dei risultati parziali, i quali si concretizzano in vari parametri che descrivono il modo in cui l'orario ha reagito alla perturbazione applicata. Dal momento che tale strumento può facilmente confrontare le tracce perturbate da quelle originariamente programmate, per ciascun treno si possono ottenere sia i ritardi primari - cioè quelli applicati direttamente dalla simulazione ad eventi discreti - che quelli secondari - causati dall'interazione con altre convogli ritardati. Avendo disponibili tali informazioni, possono essere condotte le analisi più disparate, ricavando numerosi indicatori ampliamente descritti in letteratura ([5], [15], [16]).

In questo contesto è possibile analizzare non solo la risposta di un orario in sé a varie condizioni di perturbazione, ma anche valutare l'influenza del criterio di riprogrammazione adottato. Quest'ultimo può essere adattato per modellizzare il sistema di segnalamento e controllo della linea considerata. È opportuno puntualizzare che tale soluzione modellistica, sviluppata nell'ambiente mesoscopico presentato (il quale, tra le altre cose, non considera esplicitamente le sezioni di blocco ma solo il loro effetto sul distanziamento minimo ammesso), non può sostituire pienamente una simulazione microscopica, capace di replicare il funzionamento di ogni componente del sistema ferroviario. Per questi motivi lo strumento è in grado piuttosto di simulare - con una certa approssimazione, la quale è in ogni caso quantificabile caso per caso - le operazioni di un *centro di controllo del traffico ferroviario*. L'algoritmo già presentato, con piccole modifiche, può simulare due tipi di sistemi del controllo della circolazione.

4.3.1. Sistema di controllo avanzato

È in grado di conoscere in tempo reale la posizione di tutti i treni e di comunicare loro in maniera continua la rispettiva Autorizzazione al Movimento. Questo è il caso, per esempio, dei moderni sistemi a blocco radio come ETCS-2. Tali sistemi di controllo potrebbero pienamente beneficiare da un efficace algoritmo di programmazione come quello presentato nella sezione 3.2., il quale effettua un'ottimizzazione euristica basata su un certo criterio decisionale.

In sintesi, se un'implementazione telematica sviluppata su criteri di *safety integrity* potesse permettere una "visibilità" reciproca completa e continua tra la logica di controllo (cioè l'algoritmo implementato, per esempio, all'interno dei *radio block centre*) ed il processo controllato (la circolazione effettiva), si potrebbero configurare efficaci azioni di riarrangiamento dell'orario.

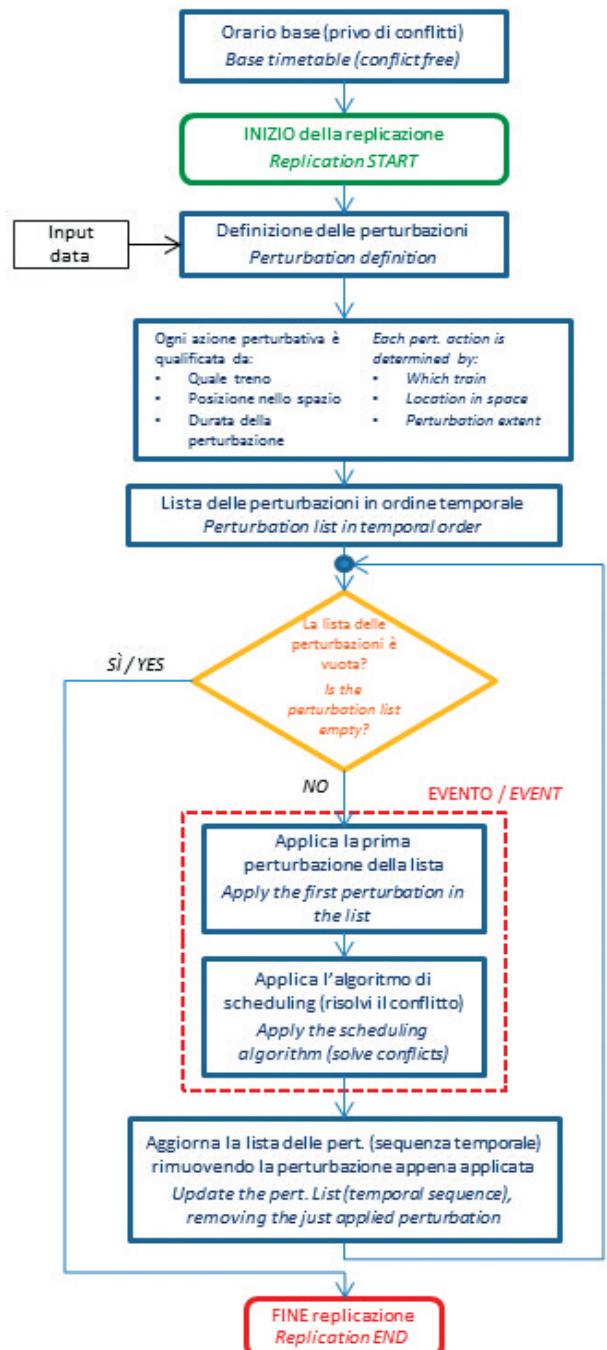


Fig. 12 - Simulazione ad eventi discreti delle perturbazioni.
Fig. 12 - Example of a saturation diagram.

computable in each particular case. The proposed algorithm, with just a few variations, can simulate two types of traffic control systems.

4.3.1. Advanced control system

It is possible to know the position of all the trains in real time as well as to communicate continuously to them

Per simulare tali sistemi, lo strumento applica l'algoritmo presentato nella sezione 3.2., con due accorgimenti. Infatti, assumendo che l'azione di riprogrammazione sia applicata non appena una perturbazione venga rilevata, le eventuali modifiche possono riguardare solo la porzione di orario successiva a tale istante. Pertanto, quando si esegue la "compressione all'indietro", viene fissato un limite inferiore prima del quale l'algoritmo non può alterare l'orario (dal momento che andrebbe a modificare eventi già avvenuti). Durante le simulazioni l'algoritmo può accelerare i treni in ritardo, riducendo nelle CS opportune il tempo di percorrenza di una certa percentuale (stirando la traccia). Questo simula il consumo degli allungamenti di percorrenza (se presenti) eventualmente introdotti precedentemente durante la fase di progetto dell'orario. Tale azione potrebbe essere imposta dal centro di controllo della circolazione come pure effettuata autonomamente dal macchinista.

4.3.2. Sistema con semplice segnalamento

Non è in grado di effettuare una reale azione di controllo della circolazione, consistente in un *dispatching* attivo e in un'alterazione dell'ordine e della sequenza dei treni. È questo il caso, per esempio, di linee in cui il traffico non è gestito da un centro di controllo (presenziato o meno), e dunque esso si basa solo sull'orario programmato e sui segnali (il cui compito è tuttavia quello di assicurare una circolazione sicura, non ottimizzata). In molti casi è comunque applicato un controllo parziale, basato con appoggio euristico su di un limitato insieme di informazioni messe a disposizione degli agenti preposti a tal compito.

Come risultato, ciascun conflitto di circolazione viene ad essere gestito dai segnali, la cui logica è però in grado di controllare l'occupazione della sezione di calcolo (in caso di conflitti di tipo 1) o di blocco (in caso di conflitti di tipo 2) protetta. Pertanto, non viene effettuata alcuna differenziazione di priorità tra i vari treni ritardati, dal momento che il primo che si avvicina ad una sezione sarà quello che la "prenoterà" prima ed occuperà poi. Per simulare questa situazione, è stata approntata una versione degradata dell'algoritmo di programmazione (fig. 13), ottenuta eliminando la possibilità di ricombinare l'orario - compressione all'indietro - e di effettuare la scelta "intelligente" del treno da penalizzare durante la soluzione di un conflitto. In tale situazione il primo treno che entra nella sezione di calcolo in esame sarà anche il primo ad uscirne (logica FIFO).

Questi due casi simulano due situazioni opposte, e possono essere utili per sviluppare un'analisi differenziale sul ruolo che la logica di riprogrammazione può ricoprire per incrementare la capacità, intesa qui come stabilità d'orario. Bisogna puntualizzare che nella maggior parte dei casi reali sono presenti situazioni intermedie, in cui è presente un limitato margine d'azione per i centri di controllo della circolazione. Essi per esempio potrebbero essere in grado di conoscere lo stato del traffico ma non

the corresponding Movement Authority. This is the case, for instance, of modern radio block systems, such as ETCS-2. These kinds of control systems could take full advantage of a performing scheduling algorithm, such as the one presented in point 3.2, which performs a heuristic optimisation based on a certain choice ratio.

In principle, if a telematics implementation, based on safety integrity concepts, allows a complete and continuous reciprocal "visibility" between the dispatching logic (i.e. the algorithm implemented, for instance, within the radio block centre) and the controlled process (the actual traffic), effective timetable re-arrangement actions could be envisaged.

In order to simulate such a system, the tool applies the algorithm presented in section 3.2., with two amendments. Assuming that the re-scheduling action is performed as soon as a perturbation arises, any modification will only concern the timetable that occurs after this instant. Therefore, when performing the "upstream compression", a low bound is set, after which it is no longer possible for the algorithm to alter the timetable (since this would modify events that have already occurred). During the simulations, the algorithm speeds up the delayed trains, by reducing the run time in the relevant CSs by a certain percentage (stressing the path). This simulates the consumption of the run time allowance (if any) previously introduced during the timetable-planning phase, commanded by the dispatching centre or operated autonomously by the driver.

4.3.2. Simple signalling system

A simple signalling system is unable to perform a real traffic control action, that is, an active dispatching of trains that would result in an alteration of their order compared to the nominal planned timetable. This is the case, for instance, of lines on which the traffic is not managed by a central dispatching centre (staffed or unstaffed), but relies only on the planned timetable and on the signals (whose task is to ensure a safe but not necessarily optimised traffic flow). In most cases, a dispatching action is actuated through a heuristic approach based on a limited set of information received by the manually operating dispatchers.

As a result, any traffic conflict would be regulated by the signals themselves, whose logic is actually only able to check the occupation of the calculation section (in the case of type 1 conflicts) or of the block section (in the case of type 2 conflicts) that each signal protects. Therefore, no priority differentiation can be made between several delayed trains, since the first one that approaches a section reserves and then occupies it. In order to simulate this situation, a degraded version of the scheduling algorithm has been set up (fig. 13), in which the possibility of rearranging the timetable (upstream compression) as well as of performing smart choices on a penalised train concerning conflict solving has been removed: the first train that enters the calcula-

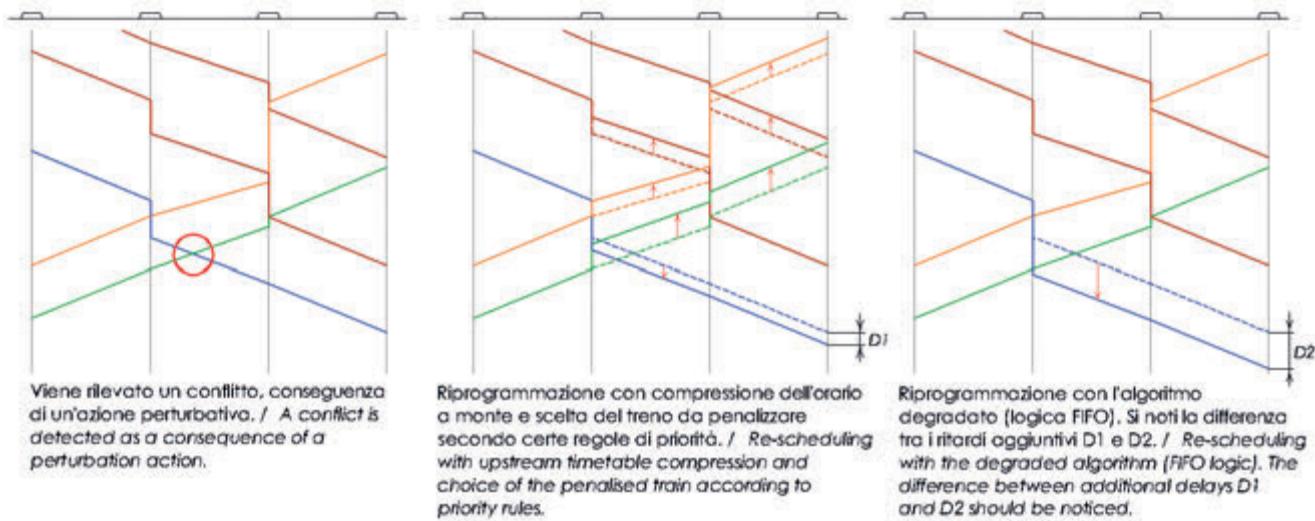


Fig. 13 - Versioni completa e degradata dell'algoritmo di riprogrammazione. Influenza sul riarrangiamento dell'orario.
Fig. 13 - Full and degraded versions of the rescheduling algorithm. Influence on timetable rearrangement.

di indicare continuamente a ciascun treno la velocità da tenere. Allo stato attuale, l'algoritmo non può modellizzare queste situazioni intermedie.

È opportuno evidenziare che l'algoritmo ad eventi discreti simula correttamente il funzionamento di un sistema di segnalamento nel caso di conflitti tra treni viaggianti in direzioni opposte su di una linea a binario singolo. Diversamente, nel caso di conflitti dovuti ad un insufficiente distanziamento temporale tra convogli rivolti nella stessa direzione, l'algoritmo trattiene uno dei due in una stazione fino a che la CS successiva non viene liberata. Ciò è irrealistico, dal momento che nella realtà il secondo treno può ripartire non appena il primo convoglio abbia liberato la prima sezione di blocco della CS: di conseguenza, nel caso di eterotachicità, il secondo treno potrebbe doversi arrestare più di una volta ai segnali di blocco intermedi presenti in quella CS. In ogni caso, la distorsione introdotta da questa discrepanza è poco significativa nella maggioranza dei casi, a parte in condizioni di severa congestione. Non si introduce peraltro nessuna approssimazione di questo tipo se la sezione di calcolo presenta un'unica sezione di blocco, come è peraltro abbastanza comune in numerose linee a singolo binario.

5. Conclusioni e sviluppi futuri

L'articolo ha presentato un algoritmo di programmazione per l'esercizio ferroviario, in grado di automatizzare il processo di costruzione degli orari. In primo luogo sono state discusse le istanze che hanno portato allo sviluppo di un metodo mesoscopico dedicato, evidenziando perché un approccio microscopico fosse inadatto allo scopo. Il beneficio principale del modello mesoscopico consiste nell'utilizzo di informazioni già aggregate per modellizzare sia l'infrastruttura che l'esercizio, riducendo

tion section of interest is also the first one to leave it (FIFO logic).

These two cases simulate two opposite extreme situations, and could be useful to develop a differential analysis on the role that the rescheduling logic could play in improving capacity, here intended as timetable stability. It should be pointed out that, in most real cases, intermediate situations exist, where only a limited re-scheduling margin is available for the dispatching centres. For instance, they might be able to know the current traffic state but are not able to continuously indicate the speed that each train should keep to. At present, the algorithm is not able to model such intermediate scenarios.

It is worth highlighting that the discrete-event algorithm is able to correctly simulate the operation of a signalling system in the case of conflicts between trains running in opposite directions on a single track. However, in the case of conflicts due to insufficient headways between consecutive trains, the algorithm keeps one of them inside a station until the subsequent CS becomes free. This is unrealistic, since in reality the second train would be allowed to depart as soon as the first train has cleared the first block section of the CS: subsequently, in the case of speed heterogeneity, the second train would have to stop several times at intermediate block signals in the CS. However, the distortion introduced by this discrepancy is not significant in most cases, except for in very congested situations. If the calculation section features only one block section - as is rather common on many single-track lines - no approximation is present.

5. Conclusions and future developments

The paper presents a scheduling algorithm for rail transport that is able to automate the timetable generation

così la quantità dei dati numerici richiesti. Per ragioni analoghe, in tale contesto anche gli algoritmi di programmazione possono essere implementati facilmente, in maniera da richiedere uno sforzo computazionale relativamente contenuto.

Sono state successivamente illustrate le implementazioni di due diversi algoritmi di programmazione, genetico uno ed euristico *greedy* l'altro. Sono state presentate le ragioni che hanno portato a preferire il secondo, evidenziando come l'algoritmo euristico *greedy* sia maggiormente controllabile durante il processo di risoluzione dei conflitti, permettendo di controllare direttamente le priorità tra i convogli.

Sono state poi presentate alcune possibili applicazioni di tale algoritmo, dedicate alla valutazione della capacità delle linee ferroviarie, e si è dimostrato come questo strumento possa essere utilizzato per ottenere in un tempo limitato dei risultati che inquadriano efficacemente il fenomeno studiato.

In particolare, nelle sezioni 4.1. e 4.2. sono descritte due applicazioni dirette dell'algoritmo di programmazione presentato; esse riguardano la generazione automatica di orari ed il relativo impiego in valutazioni di capacità. Dal momento che tale approccio richiede di analizzare numerosi orari in maniera adeguata, sono stati approfonditi gli indicatori (KPI d'orario) che possono essere monitorati e le varie maniere di interpretarli.

Infine, nella sezione 4.3. è stata presentata una diversa applicazione, in cui l'algoritmo è impiegato nella riprogrammazione degli orari per effettuare analisi di perturbazione. In questo modo è possibile simulare il sistema di segnalamento e controllo nonché l'efficacia della logica di controllo della circolazione nella gestione delle perturbazioni del traffico.

Pertanto nello stesso pacchetto integrato sono inclusi due strumenti per l'analisi di capacità sia "statica" (valutazione delle tracce orarie disponibili o programmabili per unità di tempo) sia "dinamica" (effettiva disponibilità di tali tracce in caso di perturbazioni), permettendo così un'indagine a tutto tondo. I punti di forza del modello proposto, se confrontato con altri analoghi strumenti simulativi, dovrebbero derivare dal ben calibrato compromesso tra la sua essenzialità da un lato e la sua capacità di rappresentare efficacemente l'esercizio ferroviario dall'altro. D'altro canto, la semplicità del metodo è di fatto la sua principale limitazione, rendendo l'algoritmo inadatto per una simulazione accurata delle varie entità che ne fanno parte del sistema ferroviario.

Come dichiarato precedentemente, questa ricerca è ancora in corso, e le applicazioni fin qui presentate potranno facilmente essere integrate da nuovi sviluppi. Ad ogni modo, esse rappresentano una base metodologica consolidata. In verità, non è così semplice delineare a priori nuovi approcci applicativi. D'altra parte, essi potrebbero essere immediatamente suggeriti non appena il metodo venisse applicato a casi studio oculatamente scelti, che possano

process of any train operation. The reasons that led to the development of a dedicated mesoscopic approach have been explained, and it has been highlighted why a microscopic model would be unsuitable for this task. The main benefit of the mesoscopic model is that it uses aggregated information to model both the infrastructure and the operations, thus significantly reducing the amount of required numerical data. For similar reasons, scheduling algorithms can be implemented more easily and require less computational effort.

The implementation of two different scheduling algorithms, a Genetic and a Heuristic Greedy one, has been presented. The reasons that led to the choice of the second one for further applications are discussed, and it has been pointed out how the Heuristic Greedy Algorithm is more manageable during the conflict-solving process, thus allowing trains priorities to be controlled directly.

Some possible applications of this algorithm, devoted to railway saturation evaluation, have been presented. These highlight how the proposed tool can be used to obtain results that efficiently frame the phenomenon in a rather short time.

Two direct applications of the presented scheduling algorithm have been described in detail in sections 4.1. and 4.2.; they pertain to automatic timetable generation and the related timetable-based capacity assessment. Since this approach requires several timetables to be analysed in detail, special attention has been given to the indicators (timetable KPIs) that can be monitored and to the possible ways of interpreting them.

Finally, a different application has been presented in section 4.3., where the algorithm has been employed as a rescheduling device in order to carry out perturbation analyses of an input timetable. This way, it is possible to simulate the signalling and control system and the effectiveness of the adopted traffic management logic to manage traffic perturbations.

Therefore, two tools – one for "static" (evaluation of scheduled available train paths per time unit) and one for "dynamic" capacity assessment (actual availability of such paths in the case of traffic perturbations) - are included in the same integrated package, thus allowing an all-embracing analysis. The strong points of the proposed method, compared to other railway simulation tools, derive from its well-balanced compromise between simplicity and the capability of effectively representing railway operations. However, its simplicity also represents its main limitation, as it makes the method unsuitable for an accurate simulation of the various entities involved in a railway system.

As previously mentioned, research is still ongoing and the applications here presented could easily be integrated with new ones. However, they represent a consolidated methodological basis. It is not so easy to a priori think up new application approaches. But if the method will be applied to properly chosen case studies, the need for dedicated

mettere alla prova il “raggio d’azione” dell’algoritmo. Questo potrebbe richiedere declinazioni dedicate dell’intero metodo, che potrebbero risultare sia in modifiche dell’algoritmo e dello strumento informatico sia nello sviluppo di nuove soluzioni per elaborare i risultati.

I prossimi passi di questa ricerca considereranno nell’applicare il metodo ad un significativo assortimento di casi studio, rappresentativi di diverse condizioni operative. Questa attività può anche essere utile per determinare i limiti applicativi dello strumento, evidenziandone nello stesso tempo i punti di forza. Questo porterà infine ad una metodologia completa, infusa sull’algoritmo stesso e completata da criteri guida efficaci.

declinations of the whole methodology will probably arise, and this could likely result in modifications of the algorithm and of the automatic tool as well as in the development of new approaches to post-process the results.

The next step of the research will be to apply the tool to a large variety of case studies, in order to represent different operational situations. This activity could be useful to determine the application limits of the method, but also to point out its main strong points. This activity will provide a complete methodology, based on the core algorithm, but which will also be equipped with effective application guidelines.

BIBLIOGRAFIA - REFERENCES

- [1] SACCO L., COLTRO N., DALLA CHIARA B. - *“Calculations of the carrying capacity and energy consumption on the Turin-Modane railway connection: application of models and simulation”*, Ingegneria Ferroviaria, vol. 64, issue 9, pp. 831-858, October 2011.
- [2] COVIELLO N., DALLA CHIARA B., NELLDAL B-L. - *“An assessment model of the single-track line carrying capacity: influence of the signalling system and application to the Trans-Mongolian railways”*, Ingegneria Ferroviaria, vol. LXIX issue 7-8, pp. 627-651, July-August 2014.
- [3] REITANI G., MALASPINA R. - *“La potenzialità di circolazione ferroviaria su linee a singolo binario: un modello di calcolo”*, Ingegneria Ferroviaria, vol. 50 issue 8, pp. 616-623, August 1995.
- [4] LINDFELDT O. - *“Quality on single-track railway lines with passenger traffic - Analytical model for evaluation of crossing stations and partial double-tracks”*, (Doctoral Thesis), Stockholm, Kungliga Tekniska Högskolan, pp. 1-49, 2007.
- [5] HANSEN I., PACHL J. – *“Railway Timetabling and Operations”*, 2nd edition, Eurailpress, Hamburg, Germany, ISBN 978-3-7771-0462-1, pp. 1-330, 2014.
- [6] HINDE C.J et al. - *“Train timetable generation using genetic algorithms”*, Proceedings of the International Conference on Evolutionary Computation, 2010, pp. 170-175.
- [7] NIRMALA G., RAMPRASAD D. - *“A genetic algorithm based railway scheduling model”*, International Journal of Science and Research (IJSR), vol. 3, issue 1, January 2014.
- [8] TORMOS P. et al. - *“A genetic algorithm for railway scheduling problems”*, Metaheuristics for scheduling in industrial and manufacturing applications, Studies in computational intelligence, vol. 128, pp. 255-276, 2008.
- [9] ABRIL M. et al. - *“An assessment of railway capacity”*, Transportation Research part E, vol. 44, issue 8, pp. 774-806, 2008.
- [10] CAPRARA A., FISCHETTI M., TOTH P. - *“Modeling and Solving the Train Timetabling Problem”*, Operations Research, vol. 50, issue 5, pp. 851-861, September 2002.
- [11] SCHACHTEBECK M., SCHÖBEL A. - *“To wait or not to wait and who goes first? Delay management with priority decisions”*, Transportation Science, vol. 44 issue 3 pp. 307–321, August 2010.
- [12] DOLLEVOET T., HUISMAN D. - *“Fast heuristic for delay management with passenger rerouting”*, (Internal Report), Rotterdam, Econometric Institute Report EI2011-35, pp. 1-19.
- [13] RICCI S., TIERI A. - *“Checking and forecasting of railway service regularity by a Petri Nets based simulation model”*, Ingegneria Ferroviaria, vol. LXIV issue 9, pp. 723-766, September 2009.
- [14] LARSEN R. et al. - *“Susceptibility of optimal train schedules to stochastic disturbances of process time”*, Flexible Services and Manufacturing Journal, vol. 26, issue 4, pp. 466-489, December 2014.
- [15] GOVERDE, R.M.P. et al. – *“A three-level framework for performance-based railway timetabling”*, proc. 6th international conference on railway operations, modelling and analysis, RailTokyo, Chiba Institute of Technology, 2015.
- [16] GOVERDE R.M.P., HANSEN I.A. – *“Performance Indicators for Railway Timetables”*, IEEE International Conference on Intelligent Rail Transportation (ICIRT), Beijing, pp. 301-306, 2013.
- [17] International Union of Railways (Union Internationale des Chemin de fer – UIC) (2012), Capacity. UIC code 406 R, 2nd edition, January 2012.

Sommaire

GENERATION AUTOMATIQUE D'HORAIRES ET APPLICATIONS D'ALGORITHMES DE PROGRAMMATION POUR L'ETUDE DE LA CAPACITE ET DES PERTURBATIONS DES LIGNES FERROVIAIRES

Cet article présente un instrument pour l'analyse de capacité des réseaux ferroviaires et qui se compose de deux niveaux. Un premier niveau applicatif, qui peut être considéré de type microscopique, permet de simuler numériquement la marche des trains à travers le calcul de l'occupation des sections de bloc. Un deuxième niveau, lui de type mésoscopique, emploie des données agrégées en entrée - par exemple les temps de parcours entre deux stations consécutives ou les moindres distances acceptées par le système - données qui peuvent être calculés automatiquement par le micro simulateur ou alors être rentrées manuellement par l'utilisateur.

A travers des telles données agrégées un algorithme de programmation produit des horaires faisables, optimisés davantage grâce à d'autres données paramétriques de qualité.

Le processus, étant automatisé et assez rapide au niveau du temps de calcul, peut être utilisé pour générer des ensembles d'horaires compatibles ainsi que pour produire des analyses de capacité basées sur ces horaires.

L'article présente ensuite l'implémentation du même algorithme dans un instrument dédié aux analyses de perturbation, instrument basé sur la simulation à éléments discrets, et qui applique déterminées perturbations (retards, accidents, anomalies) à un horaire de départ, lequel est reprogrammé pour résoudre les éventuels conflits de circulation. Cette simulation permet donc d'évaluer la solidité du système.

L'efficacité des algorithmes de programmation peut aussi être évaluée - algorithmes qui représentent les différentes stratégies qui peuvent être adoptées de la part des entités préposées au contrôle de la circulation.

Zusammenfassung

AUTOMATISCHE FAHRPLÄNEGESTALTUNG UND EINSATZ VON PROGRAMMBAREN ALGORITHMEN ZUR BEWERTUNG DER VERKEHRSSTORUNGEN EINER LINIE ERERGEHT VON VERKEHRS UNREGELMÄSSIGKEITEN

Es wird ein Verfahren vorgestellt, der die Verkehrsleistungen von Eisenbahnnetzwerken analysiert. Es bietet zwei Möglichkeiten an. Die erste ist der detaillierte Zulauf Simulation gewidmet und darin die Besetzungszeiten der Blockabschnitten genau bewertet werden können.

Die zweite begrenzt sich auf angliederten Daten wie, z.B., die Laufzeiten zwischen zwei anstoßenden Bahnhöfen oder die geringste Zugabstand von der Linie erlaubt. Andererseits können diese Daten mittels einem Mikrocomputer berechnet werden oder vom Betreiber auferlegt werden. Ein programmierbarer Algorithmus benutzt die obengenannten Daten um machbare Fahrpläne zu entwickeln, die nach vorbestimmten Qualitätsparametern optimiert werden können.

Das Verfahren ist automatisch und genügend schnell; es kann Menge von Fahrpläne erzeugen, die nutzbar zur Linienleistungsanalyse sind. Es wird dazu die Vervollkommen Lichtung des Algorithmus als Werkzeug zur detaillierte Analyse von Verkehrsstörungen vorgestellt. Damit die Strammheit eines Fahrplan in Beziehung von Störungen geschätzt werden kann. Außerdem kann die Wirksamkeit der verschiedenen Algorithmen evaluiert werden, was zwecks mäßig für den Verkehrskontrolle Systemen ist.

ISOTRACK. Le soluzioni che contano per il ferroviario

ISOTRACK, la divisione trasporti di **Isoil Industria S.p.A.**

dispone di una vasta gamma di strumentazione per risolvere qualsiasi problema di misura e controllo.



La nostra gamma di prodotti per il settore ferroviario comprende:

- Pick up
- Generatori e Sensori di velocità
- Sensori Radar
- Indicatori di velocità
- Registratori Statici d'Eventi (Scatola Nera)
- Display Multifunzione
- Sistemi di Videosorveglianza sui veicoli
- Misuratori di pressione, temperatura, portate e livello
- Barriere e Sensori ad infrarosso per la chiusura automatica delle porte