



Ricerca di Sistema elettrico

Sviluppo di algoritmi euristici per l'ottimizzazione del processo di elettrificazione di reti di trasporto pubblico urbano in presenza di vincoli sugli investimenti

Dario Pacciarelli, Andrea Gemma

SVILUPPO DI ALGORITMI EURISTICI PER L'OTTIMIZZAZIONE DEL PROCESSO DI ELETTRIFICAZIONE DI RETI DI TRASPORTO PUBBLICO URBANO IN PRESENZA DI VINCOLI SUGLI INVESTIMENTI

Dario Pacciarelli (Università degli Studi Roma Tre), Andrea Gemma (Università degli Studi Roma Tre)

Settembre 2017

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dello Sviluppo Economico - ENEA

Piano Annuale di Realizzazione 2016

Area: Efficienza energetica negli usi finali elettrici e risparmio di energia negli usi finali elettrici ed interazione con altri vettori elettrici

Progetto: D.7 "Mobilità elettrica sostenibile"

Obiettivo: Scenari mobilità elettrica, sub-obiettivo a.1 Strumenti di supporto TPL

Responsabile del Progetto: Antonino Genovese ENEA

Il presente documento descrive le attività di ricerca svolte all'interno dell'Accordo di collaborazione "*Sviluppo di algoritmi euristici per l'ottimizzazione del processo di elettrificazione di reti di trasporto pubblico urbano in presenza di vincoli sugli investimenti*"

Responsabile scientifico ENEA: Maria Pia Valentini

Responsabile scientifico Università degli Studi Roma Tre: Dario Pacciarelli

Indice

SOMMARIO.....	4
1 INTRODUZIONE.....	5
2 DESCRIZIONE DELLE ATTIVITÀ SVOLTE E RISULTATI.....	5
2.1 DEFINIZIONE DEI MODELLI DI PROGRAMMAZIONE MATEMATICA.....	5
2.1.1 <i>Modello 1</i>	6
2.1.2 <i>Modello 2</i>	7
2.1.3 <i>Modello 3</i>	8
2.1.4 <i>Modello 4</i>	9
2.2 RISULTATI TEORICI DI BASE PER IL PROGETTO DEGLI ALGORITMI.....	9
2.2.1 <i>Formulazione approssimata</i>	9
2.2.2 <i>Calcolo di Upper Bound e Lower Bound</i>	9
2.3 PROGETTAZIONE DEGLI ALGORITMI RISOLUTIVI EURISTICI.....	11
2.3.1 <i>Introduzione</i>	11
2.3.2 <i>Integer Knapsack Problem</i>	12
2.3.3 <i>Creazione di un set di soluzioni ammissibili attraverso le varianti del Knapsack Problem</i>	13
2.3.4 <i>Ricerca nello spazio delle soluzioni –Tabu Search</i>	13
2.3.5 <i>Integrazione tra gli algoritmi</i>	15
2.4 VALUTAZIONE DEGLI ALGORITMI EURISTICI.....	17
2.4.1 <i>Conclusioni sul confronto</i>	25
2.4.2 <i>Confronto dettagliato tra le soluzioni per l’istanza di Firenze</i>	26
2.5 IL SOFTWARE SVILUPPATO.....	39
2.5.1 <i>ONEBus</i>	39
2.5.2 <i>ONEBusBatch</i>	41
2.5.3 <i>Integrazione con i sistemi ENEA</i>	42
2.6 DESCRIZIONE DEI PRINCIPALI DATI DI INPUT DEL SW SVILUPPATO GENERATI DA BEST.....	44
3 CONCLUSIONI.....	45
4 RIFERIMENTI BIBLIOGRAFICI.....	46
5 ABBREVIAZIONI ED ACRONIMI.....	47
6 APPENDICE 1. BREVE CURRICULUM SCIENTIFICO DEL GRUPPO DI LAVORO.....	47
6.1 PROF. DARIO PACCIARELLI.....	47
6.2 PHD. ING. ANDREA GEMMA.....	48

Sommario

Il presente documento costituisce il “Rapporto tecnico descrittivo dei moduli SW e dei risultati della ricerca effettuata” dall’Università degli Studi Roma Tre prevista dall’allegato tecnico all’accordo di collaborazione per una attività di ricerca dal titolo: “Sviluppo di algoritmi euristici per l’ottimizzazione del processo di elettrificazione di reti di trasporto pubblico urbano in presenza di vincoli sugli investimenti”, stipulato tra Agenzia nazionale per le nuove tecnologie, l’energia e lo sviluppo economico sostenibile (ENEA nel seguito) e Dipartimento di Ingegneria dell’Università degli Studi Roma Tre (DI-RM3 nel seguito) nell’ambito del nell’ambito del Progetto D.7 “Mobilità elettrica sostenibile” del Piano Annuale di Realizzazione (PAR) 2016, ripartito in dieci Progetti, riferito alla seconda annualità dell’Accordo di Programma MiSE-ENEA 2015-2017. Più in dettaglio, l’attività di ricerca fa riferimento all’Area “Efficienza energetica negli usi finali elettrici e risparmio di energia negli usi finali elettrici ed interazione con altri vettori elettrici”, tematica di ricerca “Mobilità elettrica”; nello specifico, si riferisce all’obiettivo a “Scenari mobilità elettrica”, sub-obiettivo a.1 “Strumenti di supporto TPL”, del progetto “Mobilità elettrica sostenibile”.

I temi sviluppati nel suddetto accordo di collaborazione tra ENEA e DI-RM3 rientrano nell’ambito dello sviluppo di modelli di ottimizzazione in presenza di vincoli di investimento.

La ricerca del DI-RM3 ha riguardato:

1. Analisi dei dati forniti da ENEA e definizione di uno o più modelli di programmazione matematica del problema da affrontare. I modelli messi a punto hanno riguardato inizialmente due architetture: (A) *elettrificazione dei bus con ricarica al deposito* e (B) *elettrificazione dei bus con ricarica al capolinea*. La terza architettura (C) *elettrificazione dei bus con ricarica lungo il percorso*, è stata oggetto di un modello separato non ulteriormente sviluppato nell’esecuzione delle successive attività di ricerca. Per l’architettura A il modello è partito dall’assunto dell’esistenza di un unico deposito per tutta la rete e di un numero di percorsi per ogni linea maggiore o uguale a 1. Per l’architettura B il modello ha ipotizzato per ogni linea uno o più percorsi distinti tra più capilinea, su tutti i quali deve essere distribuita l’infrastruttura di ricarica necessaria al rifornimento energetico dei percorsi della linea. Per l’Architettura C il modello si è limitato ad analizzare il caso di al più due percorsi per ogni linea, uno di Andata e uno di Ritorno.
2. Progettazione di algoritmi risolutivi euristici del problema, basati sulle architetture A e B e sul rilassamento di alcuni aspetti “complicanti” del modello definito precedentemente, concordati tra DI-RM3 ed ENEA. Le semplificazioni adottate hanno consentito la risoluzione del problema in tempi estremamente contenuti, illustrati in dettaglio nella sezione dedicata alle prove sperimentali.
3. Implementazione di un dimostratore dell’algoritmo progettato ed esecuzione di test funzionali e prestazionali dell’algoritmo, sulla base delle istanze fornite da ENEA e dei risultati della procedura esatta sviluppata dall’Università di Roma Tor Vergata, che utilizza il calcolo parallelo per enumerare tutte le soluzioni del problema. I risultati della sperimentazione sono estremamente promettenti, in quanto soluzioni ottime o quasi per una rete delle dimensioni di una media città Italiana vengono sempre trovate in tempi inferiori al minuto.

1 Introduzione

Il contesto internazionale e, di conseguenza, la letteratura scientifica registrano negli ultimi anni una forte accelerazione verso quello che viene definito lo *shift to electromobility*. Il *White Paper "Roadmap to a Single European transport Area"* [1] ipotizza il *phasing out* dai veicoli a combustibile tradizionale entro il 2050. Analisi socio-economiche, spesso frutto di progetti di ricerca internazionali, si trovano nella letteratura scientifica di Europa [2,3,4,5,6], U.S.A. [7], Cina [8] e Giappone [9]. Più di recente, anche i mass-media hanno riportato di importanti decisioni di grandi aziende o nazioni in questo senso. Per esempio, a luglio 2017 l'azienda automobilistica Volvo ha annunciato che dal 2019 cesserà le attività di ricerca volte allo sviluppo di motori a sola combustione [10], mentre Emmanuel Macron negli stessi giorni ha annunciato l'intenzione della Francia di proibire la vendita di veicoli a benzina o diesel entro il 2040 [11]. La Norvegia si è spinta a prevedere l'eliminazione di tutte le auto a combustibile fossile entro il 2025 [12]. Il rapporto Bloomberg "Electric Vehicle Outlook 2017" [13] ha rivisto al rialzo le precedenti stime di crescita del mercato elettrico, portando la previsione di vendita di veicoli elettrici nel 2040 al 54% del mercato, contro la previsione precedente del 35%. Tale accelerazione è causata non solo dalla spinta ambientalista verso una riduzione delle emissioni ma soprattutto dal calo deciso previsto per il prezzo delle batterie al Litio (-70% entro il 2030, [13]). In effetti il calo dei prezzi delle batterie apre scenari del tutto nuovi anche per il trasporto pubblico, che già oggi vede la possibilità di un risparmio nei costi operativi rispetto alle tecnologie diesel o benzina.

Questo progetto intende mettere a punto uno strumento di supporto alle decisioni operative delle pubbliche amministrazioni interessate a convertire quote di trasporto pubblico dagli attuali motori diesel alle nuove tecnologie elettriche. A fronte di un budget per l'investimento iniziale in elettro-mobilità si vuole definire l'insieme di elettrificazioni più convenienti in una data rete di trasporto al fine di massimizzare la riduzione dei costi operativi. Poiché il mercato offre già numerose soluzioni che presentano costi di investimento e operativi anche molto diversi fra loro lo strumento deve non solo definire le linee interessate all'electro-mobility shift ma deve anche definire la tecnologia più conveniente per ciascuna linea da un punto di vista globale. Il progetto parte dall'identificazione di tre architetture oggi disponibili: (A) elettrificazione dei bus con ricarica al deposito, (B) elettrificazione dei bus con ricarica al capolinea, (C) elettrificazione dei bus con ricarica lungo il percorso.

Scopo specifico dell'attività di ricerca del DI-RM3 è quello di mettere a punto modelli e algoritmi di ottimizzazione rapidi ed efficienti per la soluzione di questo problema. In quest'anno di attività la ricerca si è concentrata sulle architetture A e B, con l'obiettivo di definire alcuni algoritmi euristici veloci in grado di risolvere problemi di dimensioni realistiche (media città Italiana) in tempi contenuti.

2 Descrizione delle attività svolte e risultati

2.1 Definizione dei modelli di programmazione matematica

Dall'analisi dei dati forniti da ENEA, che comprendevano la definizione di diversi modelli matematici, il DI-RM3 ha sviluppato alcuni modelli di programmazione matematica, e precisamente di Programmazione Lineare a numeri Interi (PLI) al fine di meglio definire i problemi di ottimizzazione affrontati e le loro proprietà matematiche. In questa sezione vengono illustrati i modelli prodotti, con riferimento alle diverse architetture. I primi modelli descritti nel seguito sono orientati a rappresentare correttamente una parte del problema complessivo. L'ultimo modello, descritto alla fine della sezione, è il modello base adottato per la fase successiva di sviluppo degli algoritmi.

2.1.1 Modello 1

Il primo modello sviluppato fa riferimento all'obiettivo di **elettrificare tutta la rete oggetto di studio a costo totale minimo**, utilizzando le sole architetture A e B. Questo modello, poi non ulteriormente approfondito dal progetto, era inteso come utile nella fase di sviluppo di algoritmi allo scopo di selezionare un superset promettente di capilinea da elettrificare per poi scegliere al suo interno quelli da attivare con il budget disponibile (verosimilmente inferiore al costo complessivo dell'elettrificazione di tutta la rete). Una strategia risolutiva di questo tipo ha comunque un interesse pratico in quanto consentirebbe a un'amministrazione pubblica interessata di avere una stima dei costi di elettrificazione di tutta la rete, il che consentirebbe di pianificare un piano a lunga scadenza per attuare un'elettrificazione completa. Nel seguito viene brevemente illustrato il primo modello sviluppato. Le variabili di decisione (Booleane) del problema sono indicate con le iniziali x e y , tutte le altre grandezze identificate nel seguito si riferiscono a dati e parametri del problema.

Dati e variabili del problema

Architettura A: c_{ia} = costo di elettrificazione della linea i al deposito

$x_{ia} = 1$ se si elettrifica la linea i al deposito, $x_{ia} = 0$ altrimenti.

Architettura B: $x_{ib1} = 1$ se si elettrifica la linea i solo al primo capolinea (indicato nel seguito con $j1$), $x_{ib1} = 0$ altrimenti.

$x_{ib2} = 1$ se si elettrifica la linea i solo al secondo capolinea (indicato nel seguito con $j2$), $x_{ib2} = 0$ altrimenti.

$x_{ib3} = 1$ se si elettrifica la linea i in entrambi i capolinea (indicati nel seguito con $j1$ e $j2$), $x_{ib3} = 0$ altrimenti.

y_j = numero di impianti di elettrificazione necessari per il capolinea j .

c = costo di un impianto di elettrificazione in un capolinea.

k = capacità di elettrificazione di un impianto (tempo di ricarica disponibile in un'ora, minore del reale per tener conto dei tempi di inutilizzo, fornito da ENEA)

a_{i1} = minuti di ricarica per un'ora di servizio al primo capolinea della linea i e sufficiente a fornire la carica elettrica necessaria per raggiungere il secondo capolinea

a_{i2} = minuti di ricarica per un'ora di servizio al secondo capolinea della linea i e sufficiente a fornire la carica elettrica necessaria per raggiungere il primo capolinea

$a_{i1} + a_{i2}$ = minuti di ricarica per un'ora di servizio della linea i se si elettrifica un solo capolinea

n_{linee} = numero complessivo di linee presenti nella rete in esame

$n_{capilinea}$ = numero complessivo di capilinea presenti nella rete in esame

Vincoli del problema

Elettrificazione della linea i , per $i=1...n_{linee}$:

$$x_{ia} + x_{ib1} + x_{ib2} + x_{ib3} = 1$$

Capacità necessaria al capolinea j , per $j=1...n_{capilinea}$:

$$ky_j \geq \sum_{i:j=j1(i)} [(a_{i1} + a_{i2})x_{ib1} + a_{i1}x_{ib3}] + \sum_{i:j=j2(i)} [(a_{i1} + a_{i2})x_{ib2} + a_{i2}x_{ib3}]$$

Funzione obiettivo

$$\min \sum_{j=1}^{n_{capilinea}} cy_j + \sum_{i=1}^{n_{linee}} c_{ia}x_{ia}$$

2.1.2 Modello 2

Il secondo modello sviluppato fa riferimento all'obiettivo di **elettrificare una parte della rete oggetto di studio**, utilizzando le sole architetture A e B, e con il vincolo di **rispettare il limite di budget imposto**. In questo caso la funzione obiettivo del problema di PLI è la massimizzazione del consumo energetico, che è stato l'obiettivo individuato nella prima fase della ricerca in accordo con ENEA, e successivamente modificato. Anche in questo caso, le variabili di decisione (Booleane) del problema sono indicate con le iniziali x e y, tutte le altre grandezze identificate nel seguito si riferiscono a dati e parametri del problema.

Dati e variabili del problema

Architettura A: c_{ia} = costo di elettrificazione della linea i al deposito

e_{ia} = $n_i (a_{i1} + a_{i2})$ tempo di ricarica dalla linea i se si elettrifica la linea al deposito

x_{ia} = 1 se si elettrifica la linea i al deposito

Architettura B: x_{ib1} = 1 se si elettrifica la linea i solo al primo capolinea (indicato nel seguito con j1)

x_{ib2} = 1 se si elettrifica la linea i solo al secondo capolinea (indicato nel seguito con j2)

x_{ib3} = 1 se si elettrifica la linea i in entrambi i capolinea (indicati nel seguito con j1 e j2)

y_j = numero di impianti di elettrificazione necessari per il capolinea j

c = costo di un impianto di elettrificazione in un capolinea

k = capacità di elettrificazione di un impianto (tempo di ricarica disponibile in un'ora, minore del reale per tener conto dei tempi di inutilizzo, fornito da ENEA)

a_{i1} = minuti di ricarica per un'ora di servizio al primo capolinea della linea i per arrivare al secondo capolinea

a_{i2} = minuti di ricarica per un'ora di servizio al secondo capolinea della linea i per arrivare al primo capolinea

$a_{i1} + a_{i2}$ = minuti di ricarica per un'ora di servizio della linea i se si elettrifica un solo capolinea

n_i = numero di corse AR della linea i nel periodo di osservazione (nell'ora di punta)

Vincoli

Se si elettrifica la linea i deve essere utilizzata al più una modalità, per $i=1 \dots n_{linee}$:

$$x_{ia} + x_{ib1} + x_{ib2} + x_{ib3} \leq 1$$

Capacità necessaria al capolinea j, per $j=1 \dots n_{capilinea}$:

$$ky_j \geq \sum_{i:j=j1(i)} [(a_{i1} + a_{i2})x_{ib1} + a_{i1}x_{ib3}] + \sum_{i:j=j2(i)} [(a_{i1} + a_{i2})x_{ib2} + a_{i2}x_{ib3}]$$

Vincolo di budget

$$\sum_{j=1}^{n_{capilinea}} cy_j + \sum_{i=1}^{n_{linee}} c_{ia}x_{ia} \leq budget$$

Funzione obiettivo

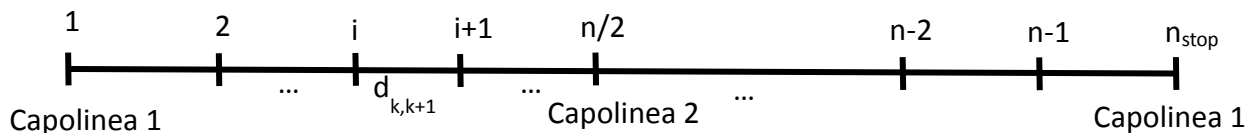
$$\max \sum_{i:j=j1(i)} [(a_{i1} + a_{i2})x_{ib1} + a_{i1}x_{ib3}] + \sum_{i:j=j2(i)} [(a_{i1} + a_{i2})x_{ib2} + a_{i2}x_{ib3}] + \sum_{i=1}^{n_{linee}} e_{ia}x_{ia}$$

2.1.3 Modello 3

Il terzo modello sviluppato fa riferimento all'obiettivo di **elettrificare una specifica linea**, utilizzando l'architettura C. L'obiettivo è quello di stabilire quali fermate debbano essere elettrificate mediante l'installazione di un impianto di ricarica, in modo tale che il costo complessivo di elettrificazione della linea si minimi. Si suppone che gli impianti abbiano capacità infinita, in quanto il tempo di ricarica di un veicolo deve essere simile al tempo necessario per la salita/discesa dei passeggeri, e quindi è verosimilmente molto inferiore al tempo di interarrivo tra due mezzi consecutivi della stessa linea. Pertanto non si pone il problema di dover mettere due impianti su una stessa fermata, anche perché tutte le fermate avranno lo stesso numero di mezzi da ricaricare.

Dati e variabili del problema

Il riferimento è una linea con uno o due capilinea, nella quale tutti i mezzi partono dal capolinea 1 e tornano al capolinea 1 dopo (eventualmente) aver stazionato presso il capolinea 2 (nel quale ciascun mezzo potrebbe effettuare una ricarica o meno). Si assume invece che nel capolinea 1 sia sempre presente una stazione di ricarica (sembra ragionevole in quanto la fermata al capolinea in genere dura di più e quindi consente ricariche migliori). Le fermate sono numerate da 1 (capolinea 1) a $n=n_{\text{stop}}$ (ancora capolinea 1) queste fermate e indico con $d_{i,i+1}$ la distanza tra la fermata i e la successiva, con $i=1\dots(n-1)$.



Le variabili di decisione (Booleane o reali) del problema sono indicate nel seguito con le iniziali x e u , tutte le altre grandezze identificate nel seguito si riferiscono a dati e parametri del problema.

$$D_i = \sum_{k=1}^{i-1} d_{k,k+1}, \text{ distanza da capolinea 1 a fermata } i \quad (D_1 = 0)$$

$$x_i = \begin{cases} 1 & \text{se si elettrifica la fermata } i \\ 0 & \text{altrimenti} \end{cases}$$

Le variabili x_i sono variabili Booleane ($i=1,\dots, n-1$) di installazione dell'impianto di ricarica NB: $x_1 = x_n = 1$

D_{max} = distanza massima tra due impianti di ricarica

$$u_i = \max_{k=1\dots i} \{D_k x_k\}$$

Le variabili u_i sono variabili reali, e registrano la distanza tra la fermata 1 e l'ultimo impianto attivato prima dello stop i (i compreso). Poiché questa espressione non è lineare e quindi non è adatta a un problema di PLI, occorre linearizzarla come segue:

Vincoli

$$u_i \geq D_k x_k \quad \text{per } k=1\dots i \text{ e per } i=2\dots n$$

$$D_i - u_i \leq D_{max} \quad \text{per } i=1\dots n$$

Il secondo gruppo di vincoli serve a imporre una distanza massima D_{max} tra due impianti di ricarica, ovvero la distanza tra lo stop i e l'ultimo impianto installato prima dello stop i deve essere inferiore alla distanza massima D_{max} .

Funzione obiettivo

$$\min 1 + \sum_{i=2}^{n-1} x_i$$

La funzione obiettivo è semplicemente la minimizzazione del numero di stazioni di ricarica, supposte tutte uguali e quindi di pari costo. Questa funzione coincide pertanto con l'obiettivo di minimizzare il costo totale di elettrificazione. Si osserva infatti che, una volta decisa l'elettrificazione della linea, tutte le altre voci di costo sono già definite (costo dei mezzi e dell'energia).

Quest'ultimo modello potrebbe essere la base per affrontare l'elettrificazione della rete includendo anche l'architettura C. Tuttavia per gli scopi della ricerca del presente accordo gli algoritmi di ottimizzazione, oggetto della prossima sezione, fanno riferimento esclusivamente alle architetture A e B. Il modello adottato in accordo con ENEA come modello di riferimento per lo sviluppo di algoritmi è il Modello 4, descritto nel seguito.

2.1.4 Modello 4

Il quarto modello sviluppato fa riferimento alla funzione obiettivo di **massimizzare il risparmio complessivo**, espresso per ogni linea come differenza tra costo della versione *to be* (a valle dell'investimento) e costo della versione *as is* (prima dell'investimento, ovvero la rete esistente). La funzione obiettivo può essere pertanto descritta come somma, su tutte le linee elettrificate, delle differenze tra i costi (di investimento e operativi) richiesti per mettere in esercizio la linea con mezzi alimentati a carburante fossile (diesel) e i costi analoghi per mezzi elettrici. In questo modello le variabili di decisione (Booleane) del problema sono indicate con le iniziali X , tutte le altre grandezze identificate nel seguito si riferiscono a dati e parametri del problema.

Per la formalizzazione matematica del problema si rimanda al Report XX predisposto da ENEA.

2.2 Risultati teorici di base per il progetto degli algoritmi

In questa sezione vengono descritti i risultati teorici ottenuti, che hanno consentito lo sviluppo degli algoritmi oggetto delle sezioni successive. Partendo dal Modello 4 descritto nella sezione precedente, è stata definita una formulazione approssimata che ha interessanti proprietà teoriche.

2.2.1 Formulazione approssimata

Nella Formulazione approssimata viene approssimato l'effetto rete sui costi di investimento ai nodi, assumendoli poco significativi rispetto al costo complessivo. Anche se questa assunzione non è sempre verificata (in alcuni casi i costi di nodo possono essere significativi), questa approssimazione è stata introdotta per ridurre la complessità del calcolo dei costi di nodo. Le conseguenze di questa semplificazione producono la formalizzazione matematica semplificata del problema illustrata nel Report XX predisposto da ENEA. Si osserva che la formulazione approssimata è comunque un problema NP-hard, per riduzione dal problema di Knapsack.

2.2.2 Calcolo di Upper Bound e Lower Bound

Il valore ottimo della funzione obiettivo della formulazione approssimata può essere sovrastimato o sottostimato risolvendo dei problemi semplificati assumendo per i coefficienti K^{na} dei valori costanti calcolati a priori. Infatti anche il valore di questi coefficienti può essere sovrastimato o sottostimato a priori

tenendo conto che se non c'è economia di scala allora $K^{na} = 1$ mentre la massima economia di scala è ottenibile elettrificando tutte le linee incidenti nel nodo n .

Nel calcolo dei bound si può quindi ignorare l'effetto rete sui costi di investimento ai nodi, assumendo che esso non determini riduzioni superiori a qualche decina di punti percentuali rispetto ai costi di nodo calcolati per le linee singolarmente prese. Tale assunzione è credibile nell'analisi dell'Architettura B, lo è meno nell'analisi dell'Arch. C, che però sarà affrontata nel prossimo anno di ricerca. Le conseguenze di questa semplificazione sono le seguenti:

- I costi di investimento e operativi al nodo sono costanti e vengono forniti dal modulo BEST per tutte le architetture $\{A,B,C\}$:
 - $[C_d^{DB}, C_d^{DT}, C_d^{DE}] = BEST_A(l)$ dove $d \in D_p, p \in P_l$
 - $[C_c^{CB}, C_c^{CT}, C_c^{CE}] = BEST_b(l)$ dove $d \in D_p, p \in P_l$
 - $[C_s^{SB}, C_s^{ST}, C_s^{SE}] = BEST_C(l)$ dove $d \in D_p, p \in P_l$
- I vincoli di massima energia consumata per le linee di tipo B e C si considerano soddisfatte dal modulo BEST
- L'allocazione ottima delle fermate per le linee di tipo C può essere ignorato.

Calcolo lower bound:

Dallo studio della singola linea si può definire un risparmio operativo fattibile R_l della linea l , dato dalla differenza tra costo di elettrificazione *stand alone* e costo tradizionale (del diesel) in una qualsiasi soluzione ammissibile, comunque generata. Indichiamo con $C(R_l)$ il costo di investimento necessario per realizzare l'elettrificazione *stand alone* della linea l i nella soluzione generata.

La funzione obiettivo diventa quindi: $\max \sum_{l \in L} R_l X_l^L$

dove $X_l^L = X_{Al}^L + X_{Bl}^L + X_{Cl}^L$

Si noti che, avendo trascurato l'effetto rete, che comporta un risparmio complessivo negli investimenti, in uno scenario in cui si elettrificano diverse linee il risparmio operativo sarà maggiore della somma dei risparmi R_l delle singole linee, mentre il costo di investimento sarà minore della somma dei costi *stand alone* $C(R_l)$. Allo stesso tempo, essendo la soluzione *stand alone* ammissibile per la singola linea, l'unico vincolo globale da soddisfare è il vincolo di budget complessivo, che diventa: $\sum_{l \in L} C(R_l) X_l^L \leq M^l$

Qualsiasi soluzione che rispetti il vincolo di budget necessariamente è anche una soluzione ammissibile del problema complessivo, in quanto il consumo di budget per l'elettrificazione dei nodi è leggermente sovrastimato. Il valore di funzione obiettivo di questa soluzione è quindi un *lower bound* (LB) del valore ottimo globale.

Si noti che l'effetto rete si può calcolare *a posteriori*. Ovvero, una volta elettrificate alcune linee, e consumata parte del budget, è possibile stimare più accuratamente l'effettivo costo di investimento. Infatti, definita una soluzione tale che $\sum_{l \in L} C(R_l) X_l^L \leq M^l$, è noto il numero delle linee elettrificate con architettura B e incidenti in un nodo e quindi si può sostituire l'espressione $\sum_{l \in L} C(R_l) X_l^L$ con il reale costo di investimento (inferiore). Questa operazione "libera" nuovo budget che può essere eventualmente utilizzato per elettrificare nuove linee, dando luogo a una procedura iterativa nella quale si attivano le linee più convenienti utilizzando i valori *stand alone* $C(R_l)$ e, successivamente, si ricalcola l'effettivo budget consumato. Il tutto viene ripetuto fino a quando non è più possibile attivare nuove linee.

Calcolo upper bound:

Un *upper bound* (UB) sull'ottimo del problema si può ottenere maggiorando opportunamente la funzione obiettivo e/o trascurando alcuni vincoli. Si noti che strutturalmente alcune componenti della funzione obiettivo sono negative o nulle, in particolare lo sono i valori:

$$(C_d^{DT} - C_d^{DE}) \leq 0; (C_c^{CT} - C_c^{CE}) \leq 0; (C_s^{ST} - C_s^{SE}) \leq 0.$$

Si può quindi maggiorare la funzione obiettivo del problema azzerando tali componenti negative e studiando quindi la funzione obiettivo semplificata:

$$\max \sum_{a \in \{A, B, C\}} \sum_{l \in L} (C_{al}^{LT} - C_{al}^{LE}) \cdot X_{al}^L$$

Allo scopo di ottenere facilmente un UB sull'ottimo è quindi possibile eliminare tutte le variabili relative all'elettrificazione del deposito e dei percorsi, ed i vincoli relativi. Restano quindi i soli vincoli:

Vincolo di Budget:

$$\sum_{l \in L} C_{Al}^{LB} \cdot X_{Al}^L + C_{Bl}^{LB} \cdot X_{Bl}^L + C_{Cl}^{LB} \cdot X_{Cl}^L \leq M^I$$

Ogni linea può essere elettrificata con un'architettura di tipo a solo se l'architettura è fattibile per la linea:

$$X_{al}^L \leq F_{al}^L \forall l \in L, \forall a \in \{A, B, C\}$$

Ogni capolinea è elettrificabile solo se sussiste la fattibilità:

$$X_c^C \leq F_c^C \forall p \in P, \forall c \in C_p$$

Ogni fermata è elettrificabile solo se sussiste la fattibilità:

$$X_s^S \leq F_s^S \forall p \in P, \forall s \in S_p$$

Una linea può essere elettrificata solo con un'architettura:

$$X_{Al}^L + X_{Bl}^L + X_{Cl}^L \leq 1 \forall l \in L$$

La soluzione ottima del rilassamento lineare di tale problema (che costituisce un UB dell'ottimo intero) si può ottenere con una semplice modifica dell'algoritmo analogo valido per il *Knapsack Problem* (KP), si veda ad esempio [14], ovvero elencando le variabili per rapporto $\frac{(C_{al}^{LT} - C_{al}^{LE})}{C_{al}^{LB}}$ decrescente e ponendo pari a 1 le variabili fino al raggiungimento del valore M^I . Fissata a 1 una delle variabili X_{al}^L vanno poste a zero le altre variabili della stessa linea l , cioè se a parità di l l'architettura con rapporto massimo è A, allora $X_{Al}^L = 1$ e $X_{Bl}^L = X_{Cl}^L = 0$.

Inoltre l'ultima variabile va posta pari al valore frazionario:

$$X_{al}^L = \frac{(M^I - \sum_{l \in L: X_{al}^L = 1} \sum_{a \in \{A, B, C\}} C_{al}^{LB} \cdot X_{al}^L)}{C_{al}^{LB}} < 1.$$

Le proprietà descritte in questa sezione consentono quindi di ottenere, sostanzialmente con degli algoritmi *greedy*, una o più soluzioni ammissibili per il problema descritto nel Modello 4. Tali soluzioni saranno tanto migliori quanto più saranno rispettate le ipotesi costruttive, ovvero quanto più i valori stand alone R_l e $C(R_l)$ saranno rappresentativi delle soluzioni con effetto rete.

2.3 Progettazione degli algoritmi risolutivi euristici

2.3.1 Introduzione

Partendo dalla teoria descritta nelle sezioni precedenti, si è deciso di implementare, in aggiunta allo schema *greedy* descritto sinteticamente nella sezione precedente, uno schema metaeuristico più sofisticato per cercare di migliorare le soluzioni iniziali prodotte dalle *greedy*. Gli schemi di riferimento

considerati, largamente usati per risolvere problemi di ottimizzazione combinatoria, vanno sotto il nome di *Iterated Local Search* o ILS [15,16] e *Tabu search*, o TS [17].

Gli algoritmi con schema ILS si basano sullo schema di ricerca locale ed evitano di bloccarsi in un minimo locale di un vicinato iterando una nuova ricerca locale su un diverso vicinato. L'algoritmo adottato nella presente ricerca, identificata una situazione di stallo con riferimento al primo vicinato, utilizza un vicinato molto ampio che viene visitato parzialmente. In particolare, viene mantenuto un pool di soluzioni che funge da base per la ricerca locale. Per ogni soluzione del pool viene prodotta una nuova soluzione ispezionando randomicamente il suo vicinato. La nuova soluzione viene costruita tenendo conto di due criteri: (i) deve essere sufficientemente diversa da quella di partenza; (ii) non deve essere eccessivamente diversa dalla precedente per non generare soluzioni "pessime". La Tabu search (TS) è invece una tecnica ideata da Fred W. Glover nel 1986 meglio definita nella sezione 3.2.4.

Partendo dalla teoria descritta nei paragrafi precedenti, come punto di partenza della ricerca ILS e TS si è deciso di utilizzare un insieme di soluzioni generate dall'applicazione di alcune varianti dell'algoritmo greedy per il knapsack problem KP descritto nella sezione precedente.

Nei prossimi paragrafi verranno ripercorse e descritte le procedure implementate e le scelte effettuate. Per permettere una facile comprensione del testo si riporteranno alcuni richiami teorici degli algoritmi usati.

2.3.2 Integer Knapsack Problem

La formulazione del problema, così come descritta nei primi paragrafi, prevede la costruzione di una soluzione soggetta a vincolo di budget a partire da un insieme di scelte, ognuna delle quali caratterizzata da un costo e da un valore. Questo tipo di formulazione può essere immediatamente tradotto nel così detto problema dello zaino: un problema di ottimizzazione combinatoria nel quale sono dati N oggetti, ognuno dei quali caratterizzato da un peso ed un valore e si vuole trovare la combinazione di oggetti che massimizza la somma dei valori, vincolando il peso ad un peso massimo sopportato senza poter dividere gli oggetti.

In letteratura questo problema è noto come Integer Knapsack Problem (IKP) e si risolve in modo esatto con algoritmi di ricerca combinatoria che richiedono ingenti risorse di calcolo. Per poter contenere i tempi di esecuzione si è deciso di applicare una versione approssimata dell'algoritmo che prevede il rilassamento del vincolo di interezza nel problema. In questo caso diviene possibile applicare un algoritmo greedy che prevede il calcolo della soluzione secondo i seguenti step:

1. Si ordinano gli oggetti in funzione della loro valore specifico o profitto $= \frac{\text{valore}}{\text{peso}}$;
2. Si sceglie l'oggetto con valore specifico maggiore che non causa il superamento del vincolo di budget e che non sia già stato inserito nella soluzione;
3. Si itera il punto 2 finché non esiste nessun oggetto da inserire che non causi il superamento del budget

Un esempio di pseudo codice viene riportato di seguito:

```
1 sBest ← s0 # soluzione
2 sWeight ← s0 # peso della soluzione
3 profit ← [] # lista dei profitti
4 for i 1 to numItems
5     profit[i]=v[i]/w[i] # per ogni item viene calcolato il profitto
6 end
7 profit.sortDescendant() # ordino i profitti in ordine decrescente
8 do
9     bestItem ← profit.pop() # estraggo l'item con maggior profitto
10    if sWeight + w[bestItem] < maxWeight # vincolo di peso
11        sBest.push(bestItem) # aggiungo l'item alla soluzione
12        sWeight ← sWeight + w[bestItem] # aggiorno il peso
```

```

13     else
14         break
15     end
16 end
17 return sBest

```

Nella formulazione specifica in analisi, l'oggetto è stato trasposto con la scelta di una tecnologia per una specifica linea, il valore è stato tradotto come valore della funzione obiettivo di una soluzione costituita dalla sola *scelta* linea-tecnologia e come peso è stato utilizzato il costo di investimento per elettrificare la linea con quella specifica tecnologia. Sostanziale differenza rispetto al knapsack tradizionale è il vincolo sulla scelta unica della tecnologia per ogni linea e quindi per ogni *scelta* effettuata vengono escluse le altre tecnologie non scelte per la medesima linea dall'insieme di scelta. L'algoritmo così costruito è un algoritmo di tipo greedy in quanto implementa la tecnica di costruzione della soluzione a partire da scelte incrementalmente ritenute di volta in volta le "migliori". L'algoritmo, al termine della propria esecuzione, fornisce una sola soluzione abbastanza buona per il problema ma per poter soddisfare la necessità di costruire un insieme di possibili soluzioni ammissibili sul quale costruire l'ILS è stato creato un set di soluzioni ammissibili attraverso 7 varianti del Knapsack Problem.

2.3.3 Creazione di un set di soluzioni ammissibili attraverso le varianti del Knapsack Problem

Per ampliare lo spazio di ricerca sono stati creati, oltre al knapsack tradizionale, 7 algoritmi simili:

1. KBM o Knapsack a budget minimo: rispetto al KP l'algoritmo termina appena viene superato il budget minimo di investimento. Questo algoritmo viene applicato ricorsivamente, generando nuove soluzioni, al variare del budget minimo da superare che viene sistematicamente impostato al budget dell'ultima soluzione calcolata;
2. KMinCost: rispetto al knapsack tradizionale le scelte non vengono ordinate per valore specifico ma per costo di investimento in ordine crescente;
3. KMaxCost: rispetto al knapsack tradizionale le scelte non vengono ordinate per valore specifico ma per costo di investimento in ordine decrescente;
4. KMinFO: rispetto al knapsack tradizionale le scelte non vengono ordinate per valore specifico ma per valore in ordine crescente;
5. KMaxFO: rispetto al knapsack tradizionale le scelte non vengono ordinate per valore specifico ma per valore in ordine decrescente;
6. KTPBest o Knapsack con tecnologie proibite: Per ogni coppia di tecnologie scelte appartenenti alla soluzione migliore tra quelle precedenti, viene generata una nuova soluzione con il knapsack tradizionale nel quale vengono proibite tali scelte.
7. KTP o Knapsack con tecnologie proibite: Per ogni soluzione generata con gli algoritmi precedenti viene iterativamente applicato l'algoritmo KP proibendo a turno una delle tecnologie scelte dalla soluzione. Questa variante è equivalente al precedente ma viene applicata a tutte le soluzioni proibendo, però, una sola tecnologia per volta.

Queste varianti non generano necessariamente soluzioni "buone" però creano una varietà della popolazione iniziale che migliora l'esplorazione globale dell'ILS.

2.3.4 Ricerca nello spazio delle soluzioni –Tabu Search

Date le N migliori soluzioni appartenenti alle soluzioni iniziali si è proceduto con lo sviluppo di un algoritmo di ricerca Tabu (TS). L'algoritmo TS è un algoritmo di ricerca locale che a partire da una soluzione iniziale effettua una ricerca locale nello spazio delle soluzioni. L'algoritmo TS tenta di superare un problema fondamentale che si presenta quando si intende realizzare un algoritmo di ricerca locale in grado di accettare mosse peggiorative ("uphill") ovvero sia l'immediato rischio di ritornare sulla precedente soluzione innescando un loop. Infatti, normalmente accade che, a partire da un punto x , se si accetta una mossa peggiorativa ad essa segue una mossa migliorativa che riporta l'algoritmo nel punto x , portando l'algoritmo ad un loop infinito. Alcuni algoritmi superano questo problema introducendo elementi di stocasticità nella ricerca della soluzione ma qualora si vogliono implementare algoritmi deterministici questo tipo di considerazione non è più valida, ed è quindi necessario porre tecniche che impediscano, o comunque rendano altamente improbabile, l'entrata in un ciclo. La più diffusa tra queste tecniche è quella delle *mosse Tabu*, che caratterizza un'ampia famiglia di algoritmi detti di ricerca Tabu (TS, da *Tabu Search*). Questi algoritmi sono, in prima approssimazione, normali algoritmi di ricerca locale, in cui cioè si compiono mosse "downhill" fin quando ciò sia possibile; quando però il punto corrente è un ottimo locale per l'intorno utilizzato, e quindi un normale algoritmo di ricerca locale terminerebbe, un algoritmo TS seleziona comunque una soluzione appartenente al vicinato, secondo criteri opportuni, e compie una mossa "uphill". Per evitare di tornare, in passo successivo, in una soluzione già visitata (in quanto migliore della soluzione corrente), l'algoritmo mantiene una lista tabu delle k mosse precedentemente effettuate, la quale impedisce di tornare in una delle soluzioni già visitate nei k passi precedenti: ad ogni iterazione, nel cercare una soluzione x con costo minore della soluzione corrente, l'algoritmo controlla la lista Tabu, e scarta tutte le soluzioni che sono generate da una mossa Tabu. Siccome l'algoritmo permette mosse "uphill", non è garantito che la soluzione corrente al momento in cui l'algoritmo termina sia la migliore determinata, si mantiene quindi, oltre alla soluzione corrente, la miglior soluzione tra tutte quelle determinate.

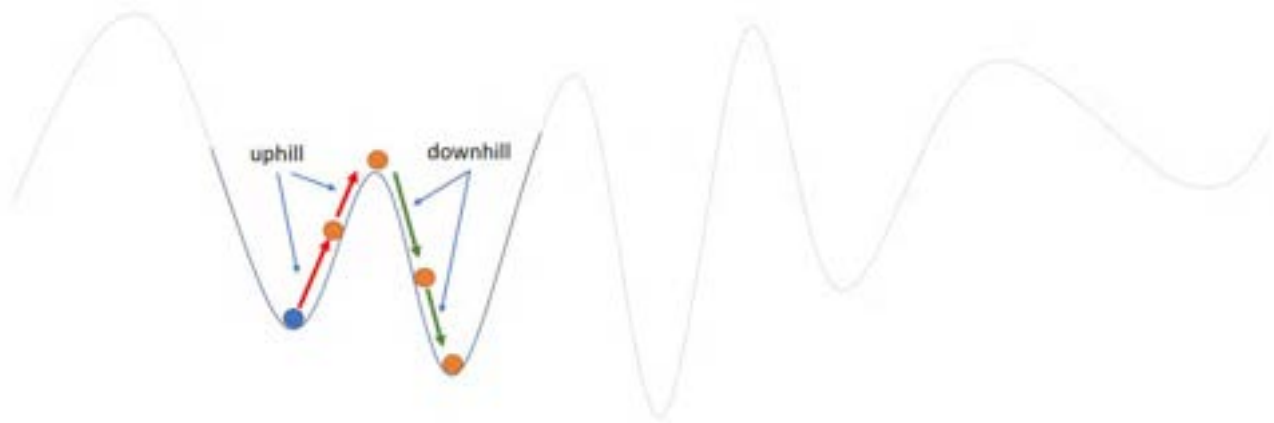


Figura 1 - Esempio di mosse uphill e downhill nella ricerca del minimo.

Un esempio di pseudo codice ¹ dell'algoritmo è:

```

1 sBest ← s0 # soluzione migliore
2 bestCandidate ← s0 # soluzione corrente
3 tabuList ← [] # Tabu list
4 tabuList.push(s0) # aggiungo la soluzione iniziale alla lista Tabu
5 while (not stoppingCondition())
6   sNeighborhood ← getNeighbors(bestCandidate) # seleziono il vicinato
7   bestCandidate ← sNeighborhood.firstElement # cerco l'elemento migliore del vicinato
a partire dalla primo elemento

```

¹ https://en.wikipedia.org/wiki/Tabu_search

```

8  for (sCandidate in sNeighborhood)
9      if ((not tabuList.contains(sCandidate)) and (fitness(sCandidate) >
fitness(bestCandidate)))
10         bestCandidate ← sCandidate
11     end
12 end
13 if (fitness(bestCandidate) > fitness(sBest))
14     sBest ← bestCandidate # se ho migliorato la soluzione aggiorno la soluzione best
15 end
16 tabuList.push(bestCandidate) # aggiorno la lista Tabu
17 if (tabuList.size > maxTabuSize) # limito la lista Tabu nelle dimensioni
18     tabuList.removeFirst()
19 end
20 end
21 return sBest

```

Esplorazione del vicinato

Una delle procedure chiave in un algoritmo TS è l'esplorazione del vicinato ovvero le mosse che vengono effettuate per trasformare la soluzione corrente, che, come abbiamo visto, possono essere migliorative o peggiorative. Le mosse applicabili nell'implementazione realizzata sono:

1. Scelta di una nuova tecnologia di elettrificazione per una linea già elettrificata;
2. Scelta di una nuova tecnologia di elettrificazione per la linea non elettrificata;
3. Eliminazione di una linea tra le linee elettrificate e sostituzione con una linea da elettrificare;

Queste mosse vengono applicate indipendentemente dal vincolo di budget, ovvero l'algoritmo può esplorare anche soluzioni non fattibili. Per evitare queste soluzioni, o almeno per stimolare la migrazione verso soluzioni fattibili, viene applicata una penale proporzionale alla violazione (cioè al superamento) del vincolo di budget, in modo da ridurre la probabilità di effettuare una mossa che lo viola. Chiaramente, a parte la soluzione corrente, che può anche essere non fattibile, l'algoritmo memorizza solo le migliori soluzioni fattibili trovate durante la ricerca e le restituisce in uscita al termine dell'esecuzione.

2.3.5 Integrazione tra gli algoritmi

Per poter effettuare una ricerca globale, e restituire in uscita un insieme di soluzioni, si è deciso di implementare uno schema di ricerca *multi-start* attraverso due procedure:

- Parallelizzazione della ricerca a partire da più soluzioni iniziali ottenuti dalle greedy;
- Generazione di soluzioni pseudo-casuali di ripartenza per la TS.

La prima delle due procedure consiste nell'applicare un algoritmo TS ad ogni soluzione generata dagli algoritmi di knapsack in modo da esplorare più parti dello spazio delle soluzioni.

Quando uno di questi algoritmi non riesce a trovare una soluzione migliorativa o fattibile per un numero prefissato di mosse allora, per lo specifico TS, viene identificata una situazione di stallo. A questo punto viene generata una nuova soluzione a partire dalla soluzione attuale del TS, facendo ripartire l'algoritmo da questa nuova soluzione. La procedura di generazione casuale delle soluzioni, affiancata alla paralizzazione delle ricerche locali permettono l'esplorazione parallela di diverse zone dello spazio di ricerca.

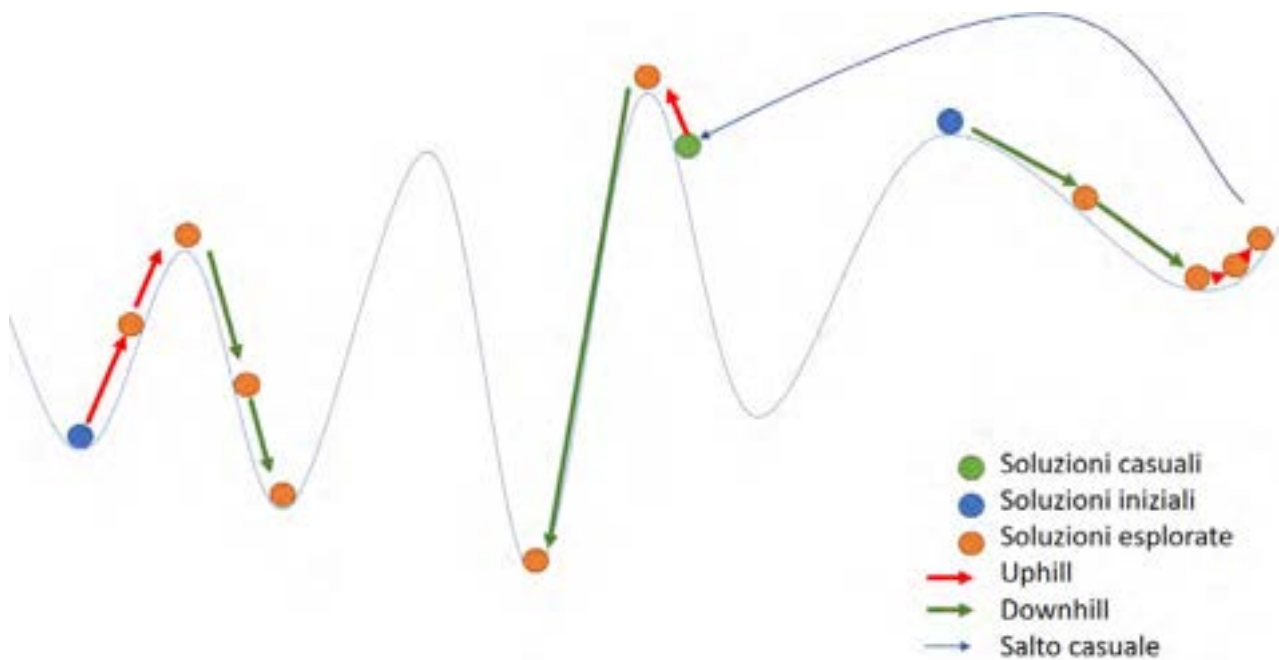


Figura 2 - Esempio di esplorazione parallela dello spazio delle soluzioni

Per limitare i tempi di calcolo e l’impatto sulle risorse dell’elaboratore, si pongono apriori i seguenti limiti:

1. Numero massimo di TS paralleli;
2. Numero massimo di soluzioni esplorate complessivamente;
3. Numero di iterazioni necessarie all’identificazione dello stallo;

L’algoritmo mantiene comunque in memoria le soluzioni migliori trovate da tutti gli algoritmi per poter fornire l’output desiderato all’utente.

In Figura 3 è possibile avere un quadro complessivo del collegamento esistente tra i diversi algoritmi rappresentativo del funzionamento dello schema ILS. In particolare vengono eseguiti il Knapsack tradizionale (KP) e le sue varianti basate sul costo e sulla funzione obiettivo: KMinCost, KMaxCost, KMinFo, KMaxFo e KBM. Quest’ultimo si applica alla soluzione del KP e si itera fino a quando produce nuove soluzioni. Alla migliore soluzione trovata viene applicato l’algoritmo KTPBest che inibisce sistematicamente a coppie le tecnologie scelte per la soluzione. Infine a tutte le soluzioni viene applicato il KTP che sistematicamente inibisce una tecnologia tra quelle scelte nella soluzione.

A tutte le soluzioni così generate viene applicato un algoritmo TS che può essere rilanciato su di una nuova soluzione qualora si identifichi una situazione di stallo. Durante tutta l’esecuzione l’algoritmo archivia le migliori soluzioni trovate che vengono riportate come output all’utente una volta terminato il numero massimo di soluzioni esplorabili.

L’integrazione di tutti gli algoritmi verrà chiamato, nell’ambito di questo documento, KTS.

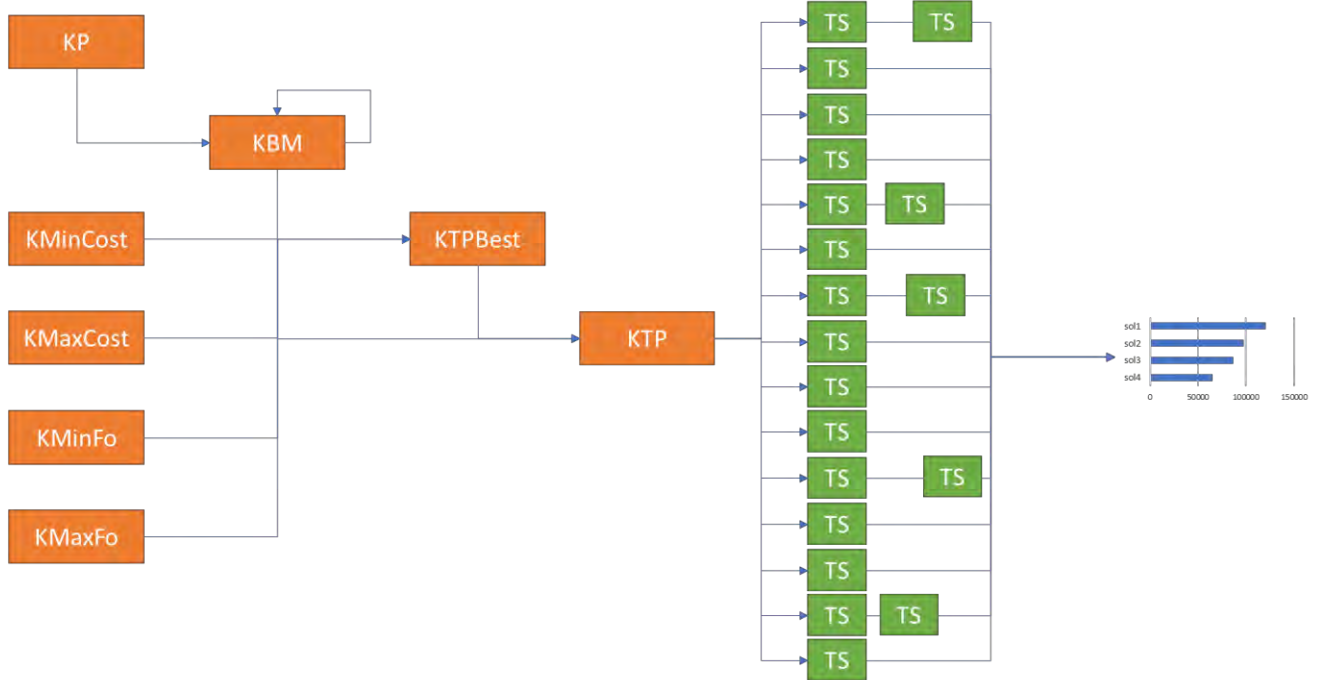


Figura 3 - Quadro complessivo dell'algoritmo di ottimizzazione

2.4 Valutazione degli algoritmi euristici

L'algoritmo KTS è stato confrontato, per una sua validazione, con l'algoritmo esatto sviluppato, nell'ambito del medesimo progetto, dall'Università di Tor Vergata. Quest'ultimo algoritmo, utilizzando il sistema di super calcolo CRESCO ha esplorato l'insieme completo delle soluzioni alla ricerca della soluzione ottima. Per poter confrontare i due algoritmi sono state create delle istanze ad hoc sulle quali è stata valutata la capacità dell'algoritmo approssimato KTS di avvicinarsi alla soluzione ottima. I parametri che hanno caratterizzato ogni istanza elaborata sono state:

- N° di linee: caratterizzante la dimensione dello spazio delle soluzioni;
- N° di capolinea: caratterizzante il livello di sovrapposizione della rete;
- Budget: caratterizzante lo spazio delle soluzioni ammissibili e utilizzato per il pruning durante l'esecuzione dell'algoritmo esatto dell'UTV.
- Tolleranza: Rapporto tra budget minimo ammissibile e budget.
- Seed: seme iniziale del generatore casuale di numeri utilizzato nella creazione della rete di test.

In riferimento ai costi generati, ogni voce di costo è stata definita positiva ed è un valore estratto casualmente tra 1 e 9: Questa scelta ha creato soluzioni con FO per lo più negative.

Le istanze create sono state generate usando i seguenti parametri:

- 30 linee;
- 6,10 o 20 capolinea;
- 30, 60 o 90 di budget;
- Tolleranza 0.8;
- 5 seed differenti per ciascuna istanza;

La codifica data alle istanze è la seguente: l{#linee}_c{#capolinea}_b{budget}_t{tolleranza}_{seed} da cui l30_c10_b90_t0.8_3 sta ad indicare un'istanza con 30 linee, 10 capolinea, 90 di budget, 0.8 di tolleranza e seed 3.

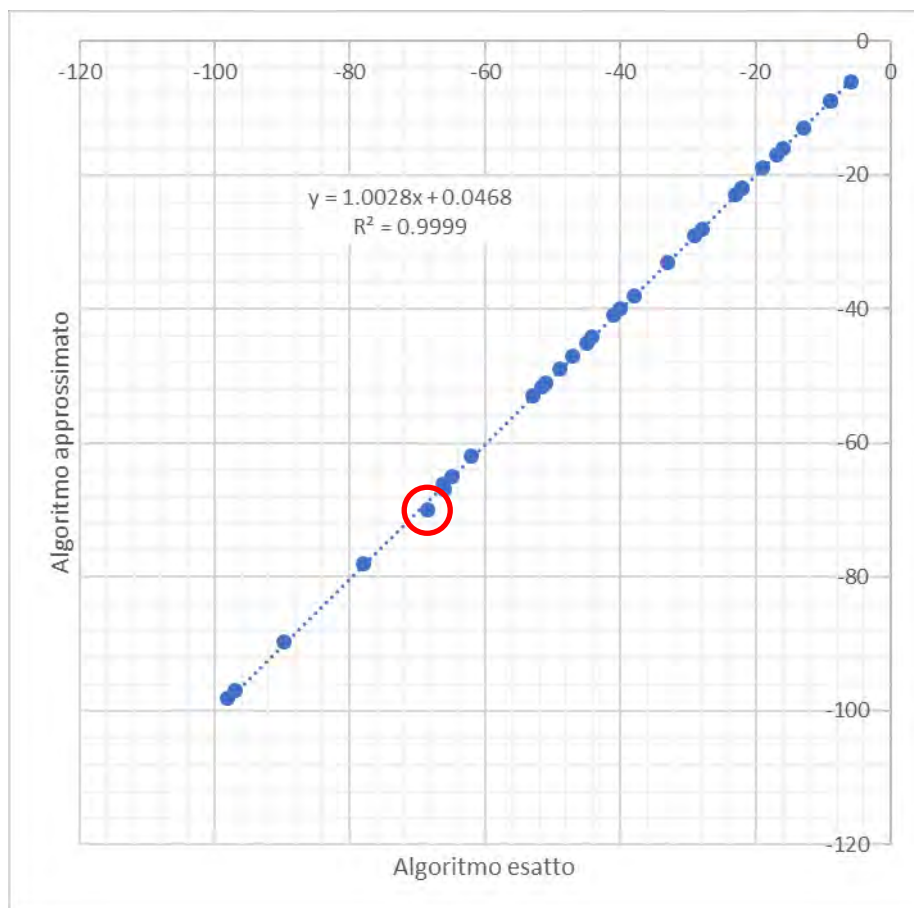


Figura 4 - Confronto tra le soluzioni migliori delle singole istanze

Dal confronto dei risultati dei due algoritmi alle istanze di prova è emerso che l'algoritmo approssimato trova sempre la soluzione ottima tranne che per un caso, dove la differenza si mostra essere minima. (istanza: l30_c10_b90_t0.8_3).

Andando ad effettuare il medesimo confronto sulle prime 5 soluzioni trovate per ogni algoritmo si ottiene lo scattergram presente in Figura 5 da cui si può notare come ci sia una sostanziale similitudine dei due set di soluzioni. Ovviamente l'algoritmo approssimato effettua una leggera sottostima, che nel 3° quadrante del diagramma cartesiano corrisponde ad un coefficiente angolare > 1 per la linea di tendenza.

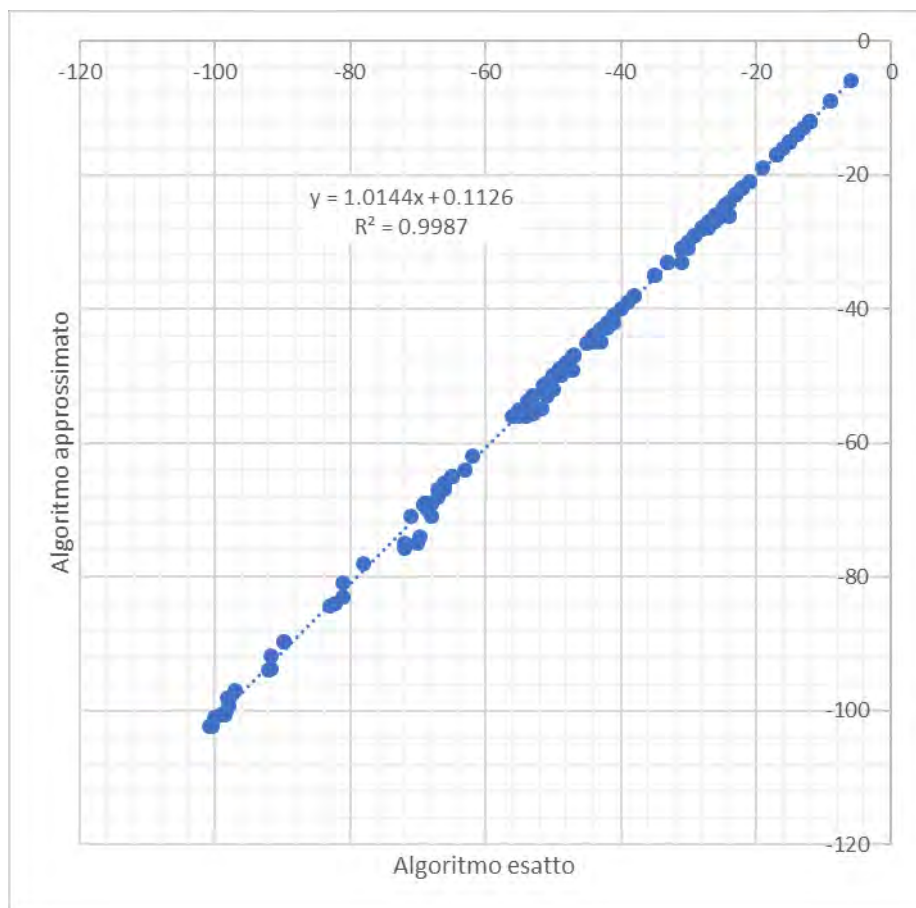


Figura 5 - Confronto tra le prime 5 soluzioni dell'algoritmo esatto e le prime 5 di quello approssimato

Confrontando le singole istanze emerge una sostanziale similitudine delle prime soluzioni di ogni istanza, uguaglianza che tende a diminuire nelle soluzioni successive. Di seguito alcuni esempi:

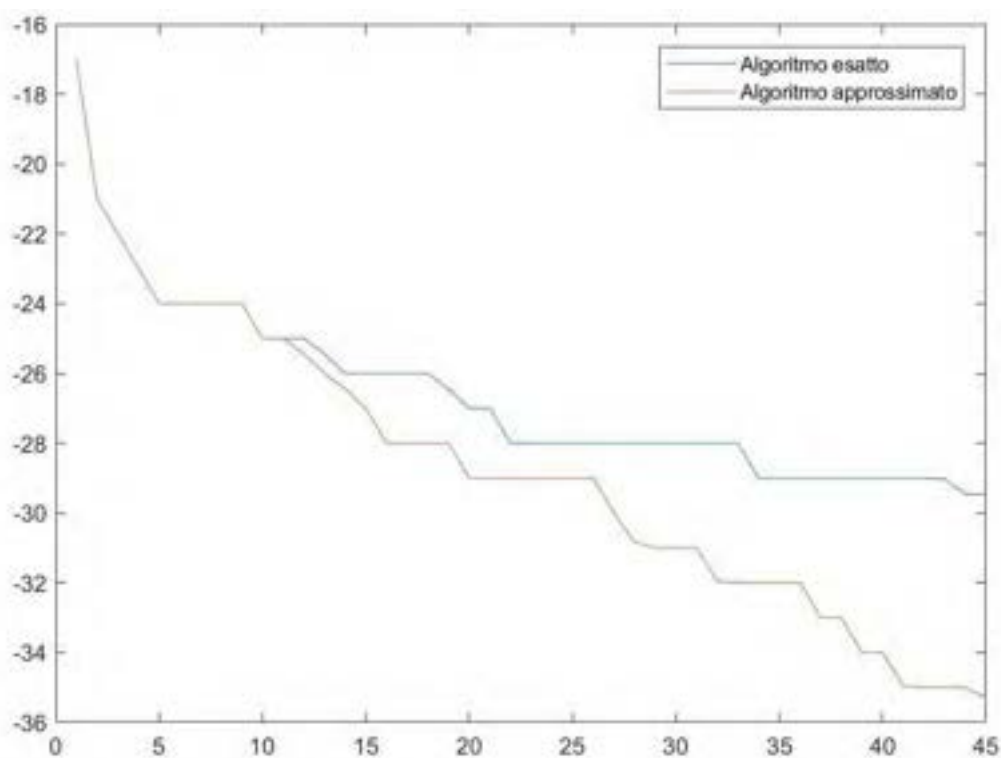


Figura 6 - Confronto tra le FO dei 2 algoritmi per le prime 45 soluzioni (I30_c10_b30_t0.8_3)

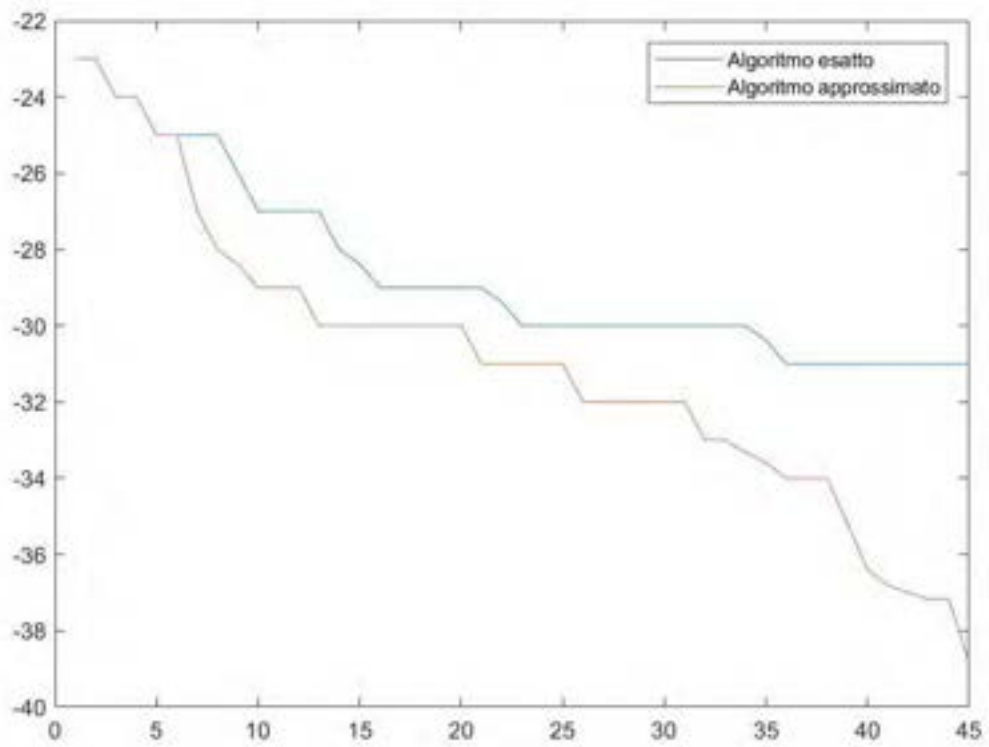


Figura 7 - Confronto tra le FO dei 2 algoritmi per le prime 45 soluzioni (I30_c20_b30_t0.8_3)

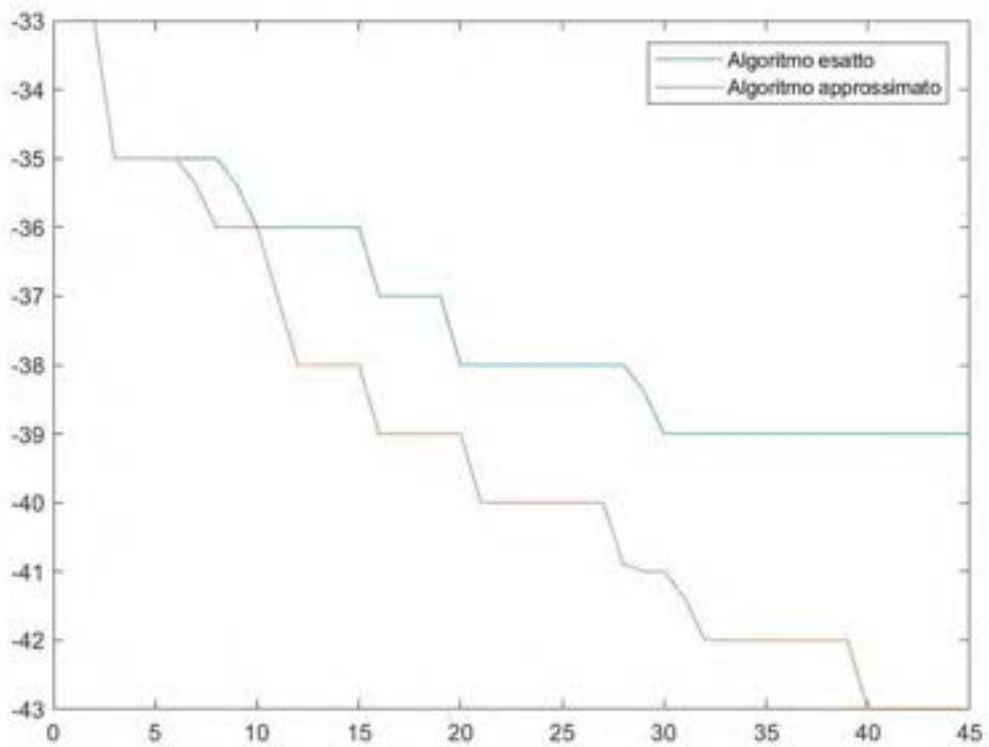


Figura 8 - Confronto tra le FO dei 2 algoritmi per le prime 45 soluzioni (I30_c10_b60_t0.8_3)

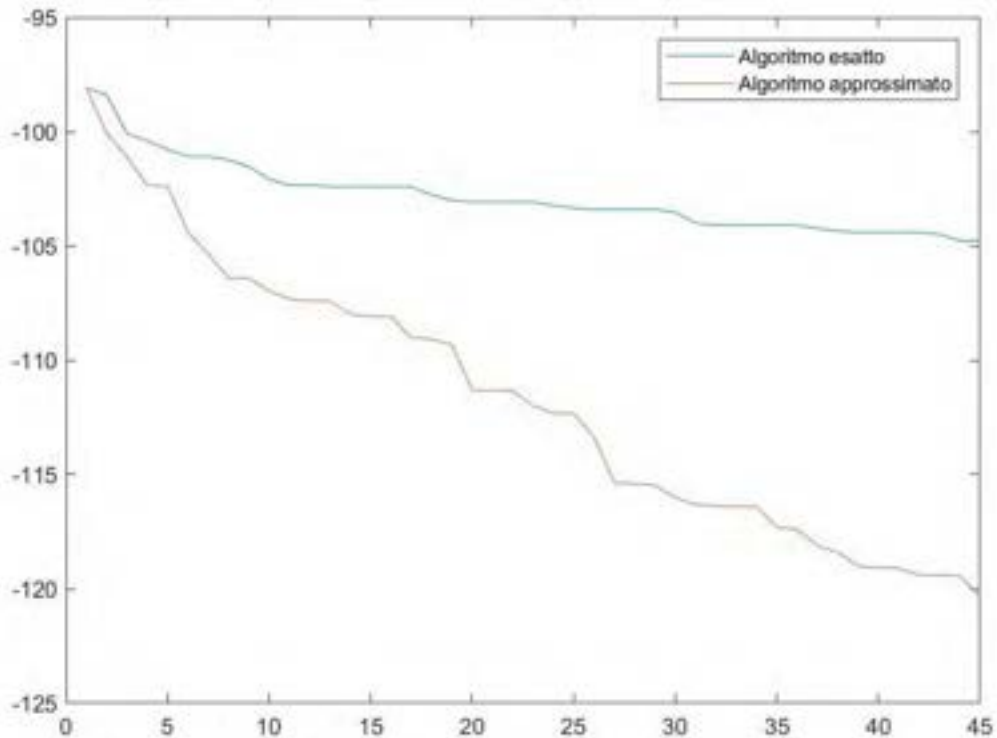


Figura 9- Confronto tra le FO dei 2 algoritmi per le prime 45 soluzioni (I30_c6_b90_t0.8_3)

Per tutte le istanze analizzate è stata anche effettuata una rappresentazione grafica delle soluzioni per evidenziare le differenze all'interno del set di soluzioni non solo in termini di FO ma anche in termini di tecnologie adottate: alcuni esempi sono visibili in Figura 10, Figura 11, Figura 12 e Figura 13. In quasi tutti i casi, si evidenzia una sostanziale similitudine tra le prime n soluzioni dell'algoritmo esatto che potrebbe andare ad indicare la presenza di ottimi locali nell'intorno della soluzione ottima e quindi zone di quasi-convessità della funzione obiettivo. Queste aree di convessità sicuramente hanno contribuito agli ottimi risultati dell'algoritmo approssimato.

Analogamente sono state rappresentate le soluzioni dell'algoritmo approssimato che presentano, come era lecito aspettarsi, delle similitudini interne, verso le soluzioni dello stesso algoritmo, e similitudini verso le soluzioni dell'algoritmo esatto. Da queste rappresentazioni si deduce la capacità, dell'algoritmo approssimato, di esplorare lo spazio delle soluzioni in zone lontano dalla soluzione ottima. Infatti, anche se numericamente inferiori, nel set di soluzioni buone sono presenti alcune soluzioni lontane della soluzione ottima e dalla sua area di convessità.

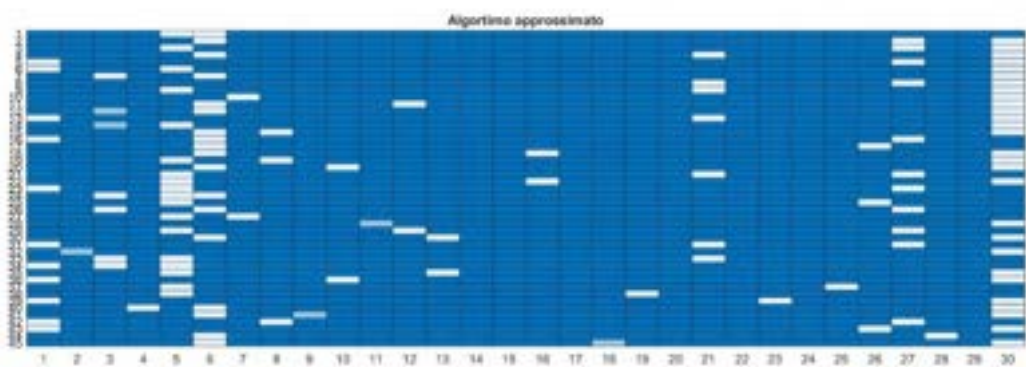
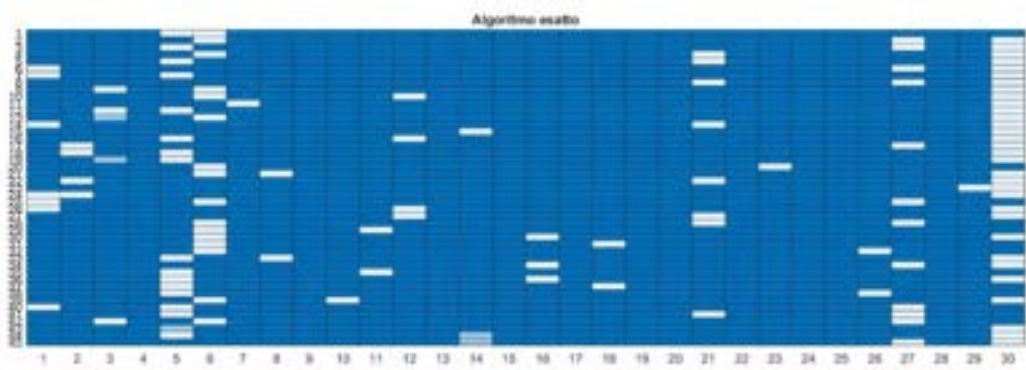


Figura 10 - Soluzioni a confronto tra l'algoritmo esatto e l'algoritmo approssimato. Il blu identifica l'assenza di elettrificazione il bianco identifica la tecnologia A e il celeste la tecnologia B (I30_c10_b30_t0.8_3)

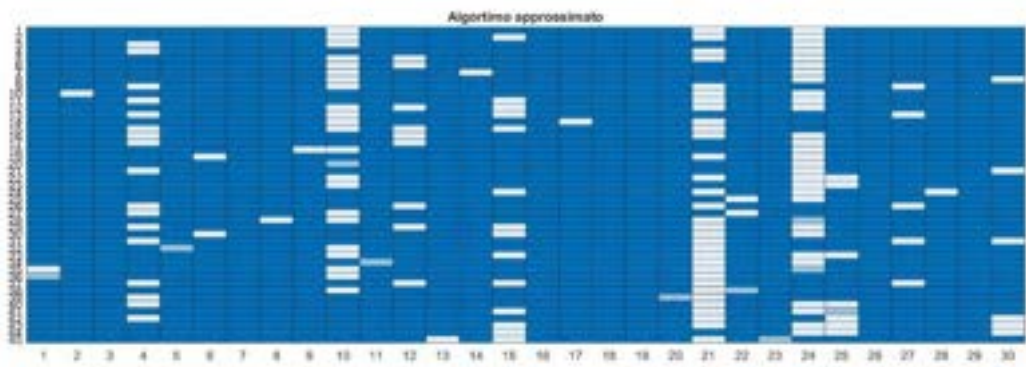
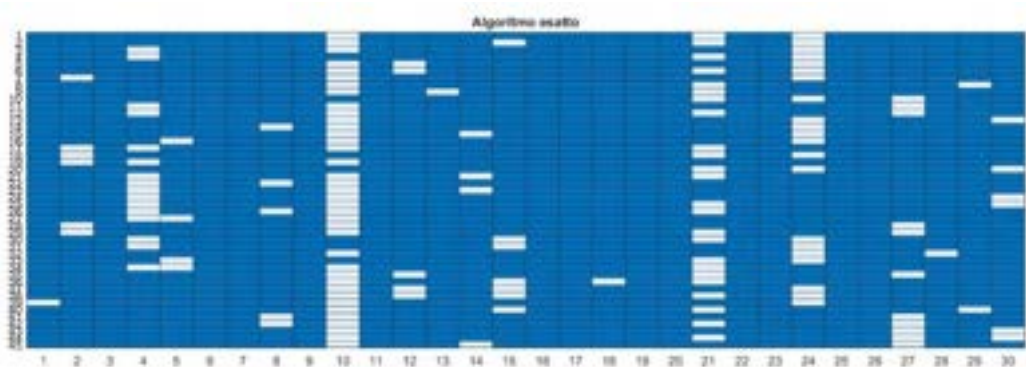


Figura 11-- Soluzioni a confronto tra l'algoritmo esatto e l'algoritmo approssimato. Il blu identifica l'assenza di elettrificazione il bianco identifica la tecnologia A e il celeste la tecnologia B (I30_c20_b30_t0.8_3)

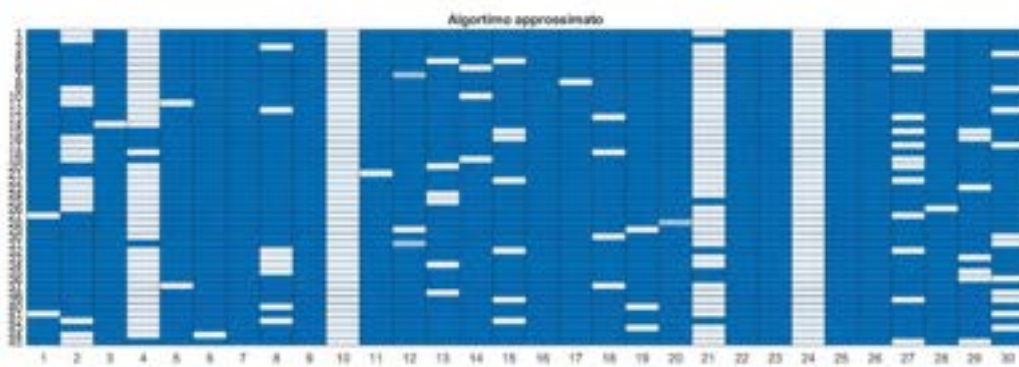
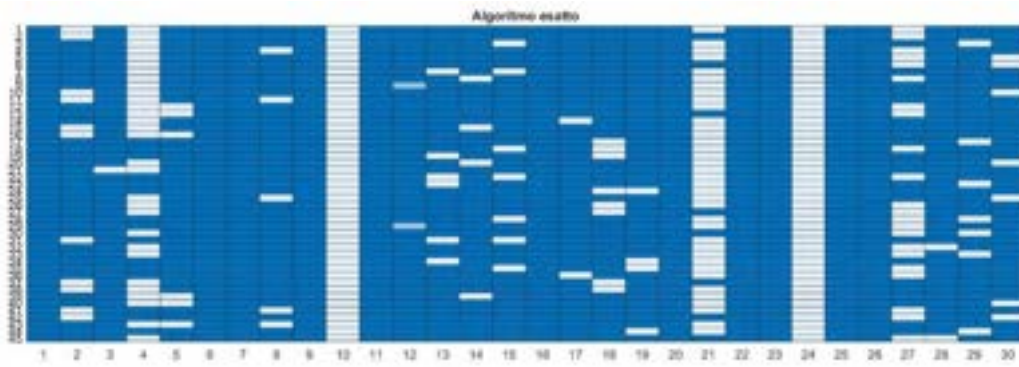


Figura 12 - Soluzioni a confronto tra l'algoritmo esatto e l'algoritmo approssimato. Il blu identifica l'assenza di elettrificazione il bianco identifica la tecnologia A e il celeste la tecnologia B (I30_c10_b60_t0.8_3)

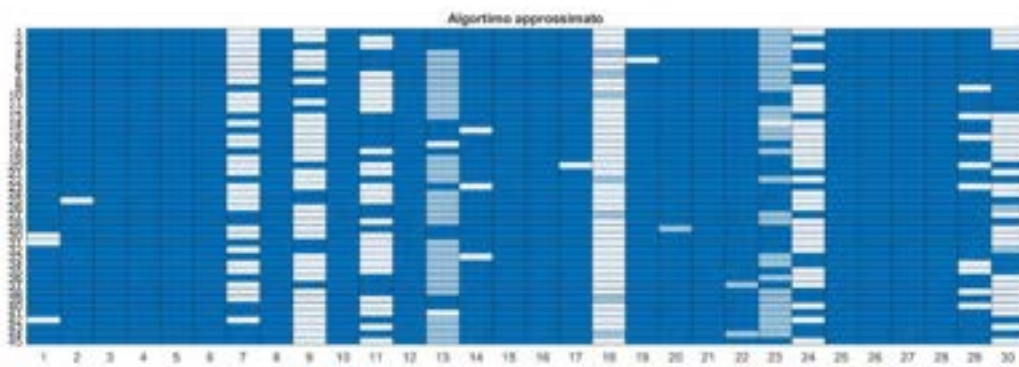
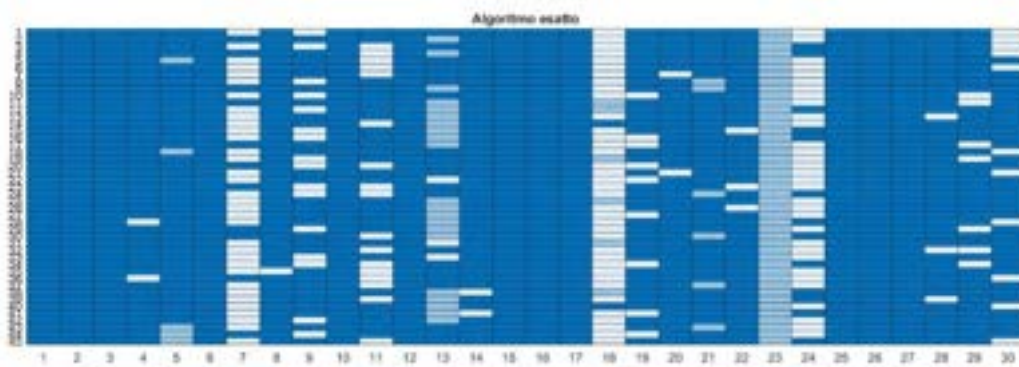


Figura 13 - Soluzioni a confronto tra l'algoritmo esatto e l'algoritmo approssimato. Il blu identifica l'assenza di elettrificazione il bianco identifica la tecnologia A e il celeste la tecnologia B (I30_c10_b90_t0.8_3)

Analoghe analisi sono state effettuate su alcune istanze di Firenze caratterizzate da un budget relativamente ridotto. In particolare le istanze valutate sono state:

- Budget 2.000.000€ e tolleranza 0.9: l85_c101_b2000000_t0.9_1
- Budget 2.000.000€ e tolleranza 0.9: l85_c101_b2000000_t0.9_1
- Budget 4.000.000€ e tolleranza 0.1: l85_c101_b4000000_t0.1_1
- Budget 4.000.000€ e tolleranza 0.9: l85_c101_b4000000_t0.9_1
- Budget 5.000.000€ e tolleranza 0.1: l85_c101_b5000000_t0.1_1
- Budget 5.000.000€ e tolleranza 0.9: l85_c101_b5000000_t0.9_1
- Budget 6.000.000€ e tolleranza 0.1: l85_c101_b6000000_t0.1_1
- Budget 6.000.000€ e tolleranza 0.9: l85_c101_b6000000_t0.9_1
- Budget 7.000.000€ e tolleranza 0.9: l85_c101_b7000000_t0.9_1

In questo caso dal confronto tra i due algoritmi emerge che l'algoritmo approssimato riesce sempre a trovare la soluzione migliore e ad ottenere "buoni" risultati anche nella ricerca delle prime 5 soluzioni.

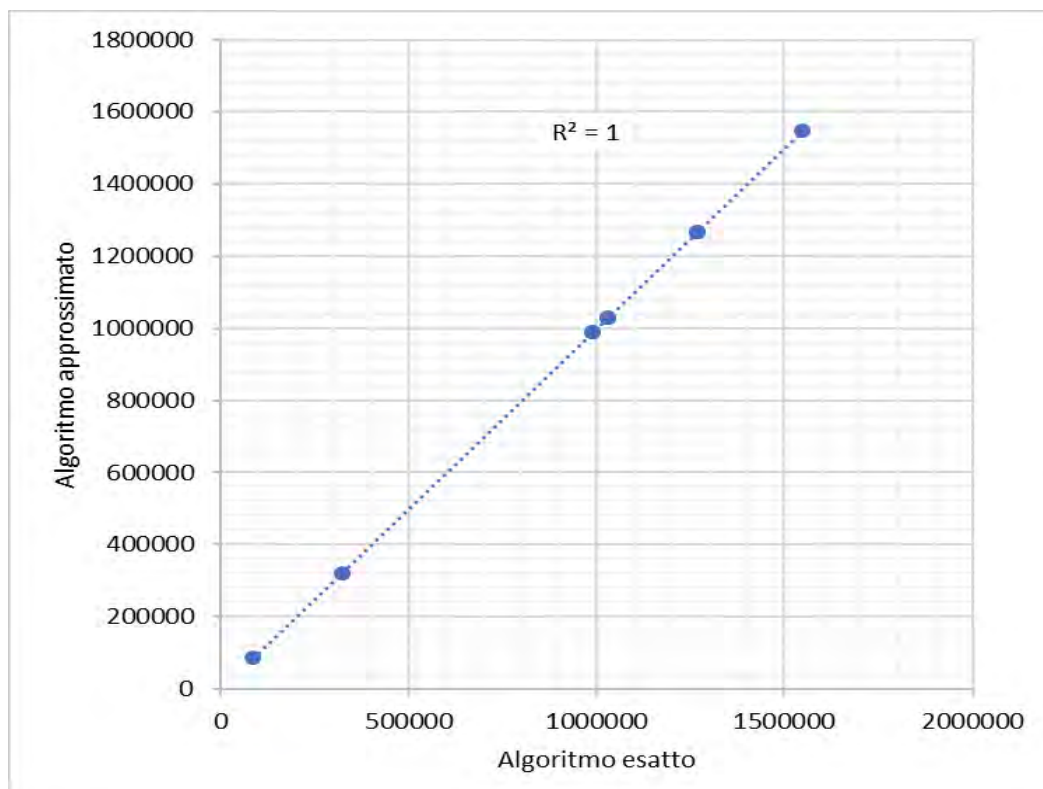


Figura 14 Confronto tra le soluzioni migliori delle singole istanze dei 2 algoritmi

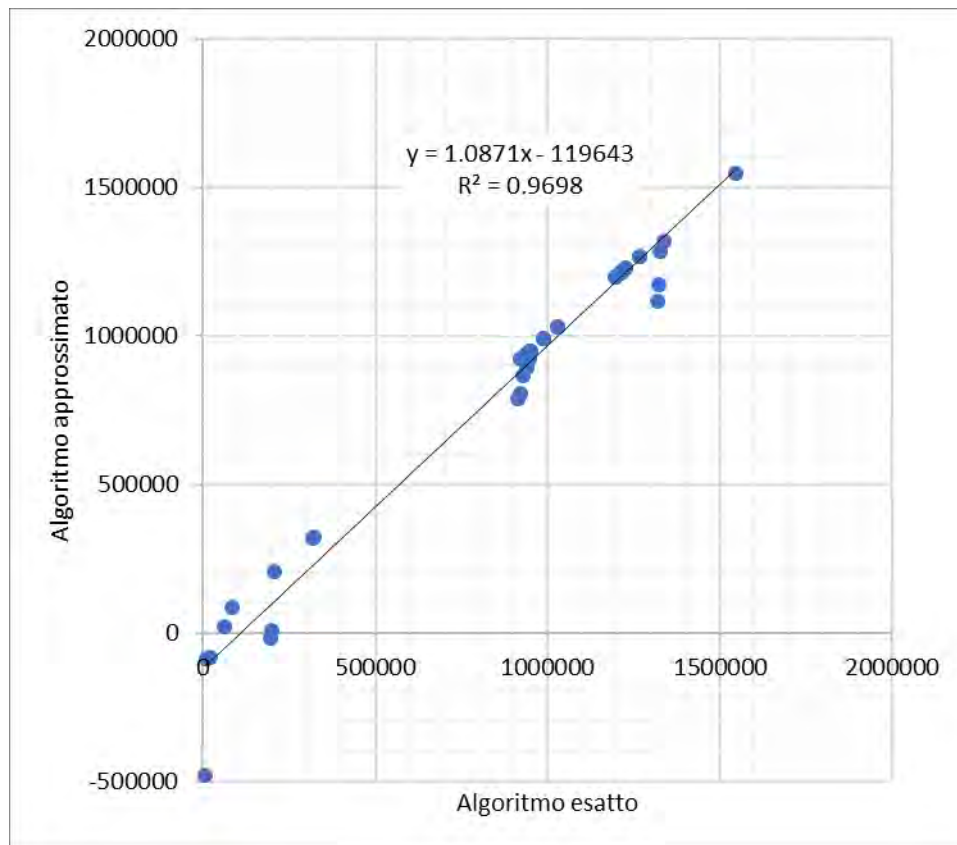


Figura 15 - Confronto tra le prime 5 soluzioni dell'algoritmo esatto e le prime 5 di quello approssimato

2.4.1 Conclusioni sul confronto

L'algoritmo approssimato KTS ha mostrato ottimi risultati nella ricerca della soluzione ottima, trovandola in tutte le istanze tranne che in una. Ottimi risultati sono stati trovati anche nell'esplorazione dello spazio delle soluzioni per la ricerca di un insieme ammissibile di soluzioni buone.

L'algoritmo ha lavorato egregiamente sia sottoposto a vincoli stringenti (budget e tolleranza) che su istanze reali come la rete di Firenze. Il numero di soluzioni esplorate dal TS è stato sempre limitato a 10.000 mantenendo i tempi di esecuzione limitati a pochi minuti e dipendenti solo dalla dimensione dell'istanza.

Durante l'analisi effettuata si è potuto appurare che un buon incremento dell'efficacia dell'algoritmo si è ottenuto attraverso l'utilizzo del Knapsack e delle sue varianti per il calcolo delle soluzioni iniziali.

Gli algoritmi TS contribuiscono notevolmente all'esplorazione dello spazio delle soluzioni migliorando la FO media del set di soluzioni migliori.

2.4.2 Confronto dettagliato tra le soluzioni per l'istanza di Firenze

In riferimento a tutte le istanze su Firenze vengono riportati l'andamento delle FO, la rappresentazione grafica e numerica di tutte le soluzioni.

Da queste si deduce che il comportamento di similitudine tra i due algoritmi si manifesta anche sulle istanze reali di Firenze ma che il vincolo di budget riduce di molto la capacità dell'algoritmo approssimato di esplorare efficacemente lo spazio delle soluzioni.

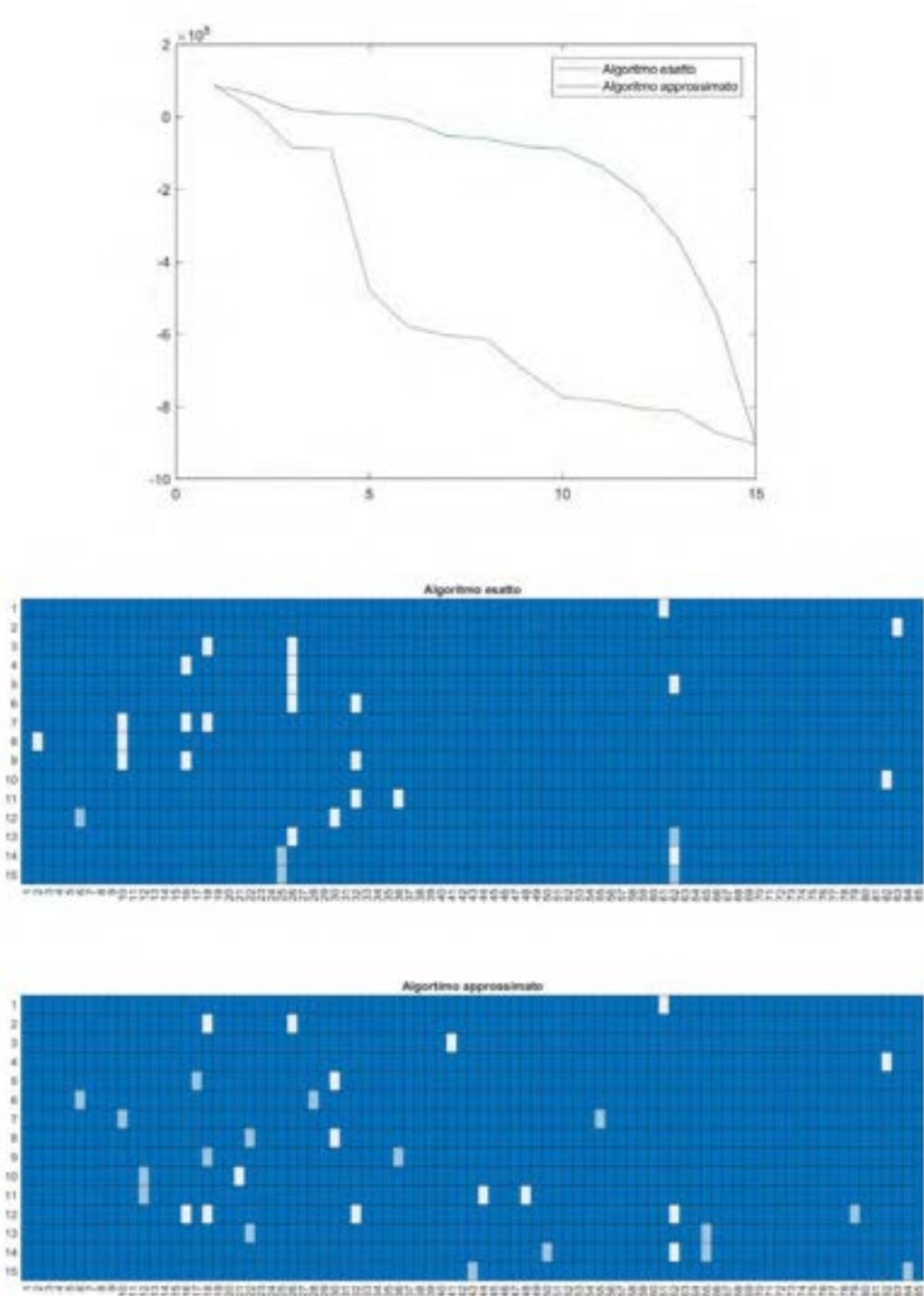


Figura 16 -Soluzioni a confronto tra l'algoritmo esatto e l'algoritmo approssimato. Andamento della FO per le differenti soluzioni e rappresentazione della soluzione attraverso una mappa di calore: Il blu scuro = Diesel,

bianco = Tecnologia A, celeste = tecnologia B
[Istanza: l85_c101_b2000000_t0.9_1](#)

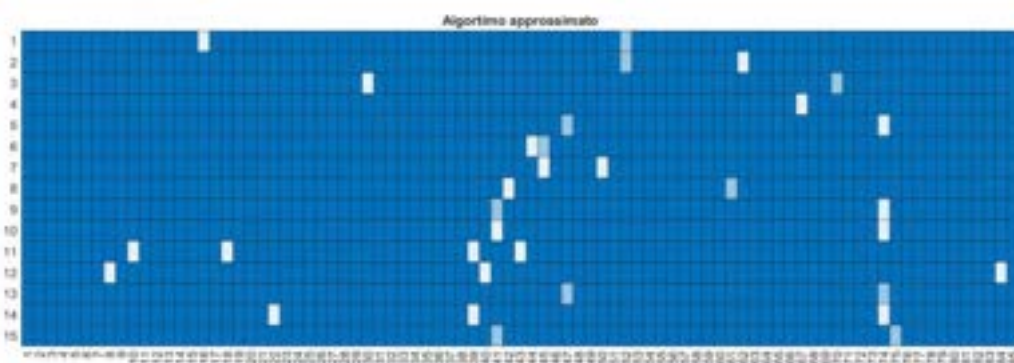
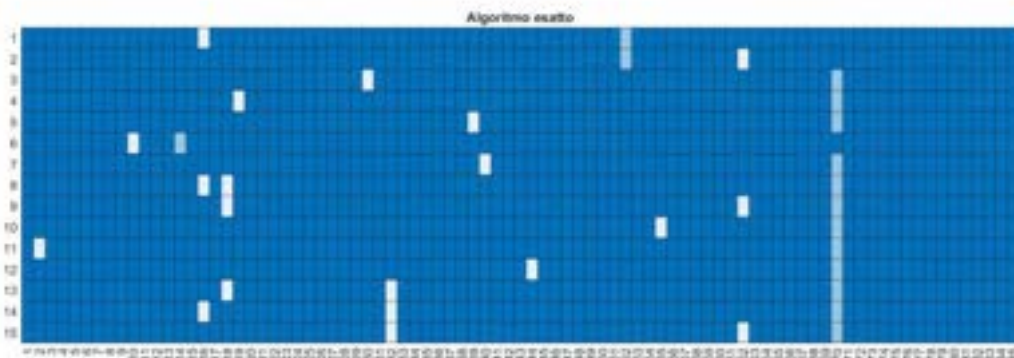
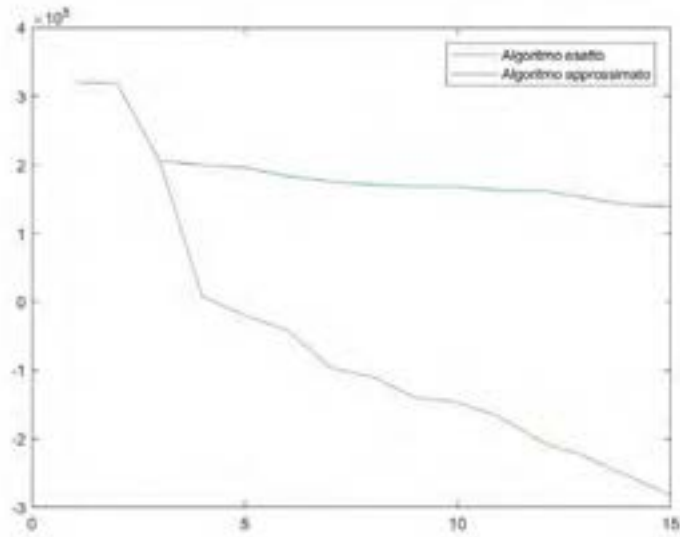


Figura 17 - Soluzioni a confronto tra l'algoritmo esatto e l'algoritmo approssimato. Andamento della FO per le differenti soluzioni e rappresentazione della soluzione attraverso una mappa di calore: Il blu scuro = Diesel, bianco = Tecnologia A, celeste = tecnologia B
[Istanza: l85_c101_b3000000_t0.9_1](#)

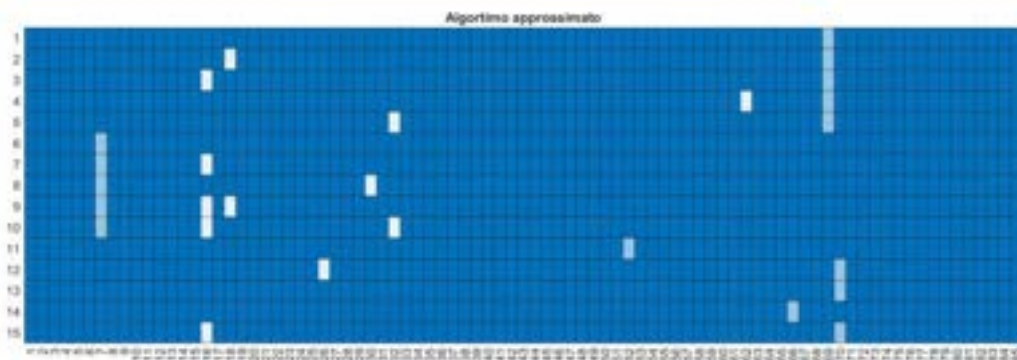
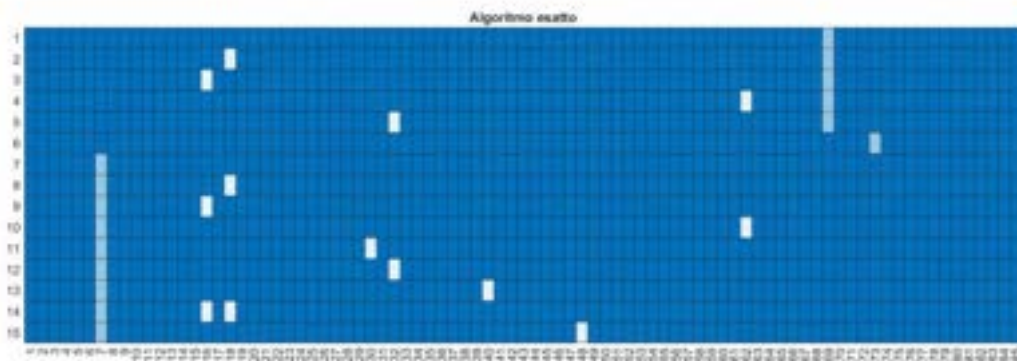
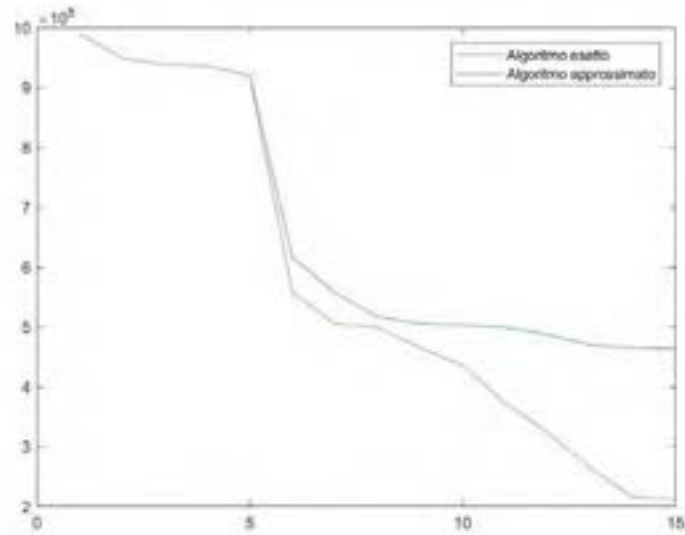


Figura 18 - Soluzioni a confronto tra l'algoritmo esatto e l'algoritmo approssimato. Andamento della FO per le differenti soluzioni e rappresentazione della soluzione attraverso una mappa di calore: Il blu scuro = Diesel, bianco = Tecnologia A, celeste = tecnologia B
 Istanza: [l85_c101_b4000000_t0.1_1](#)

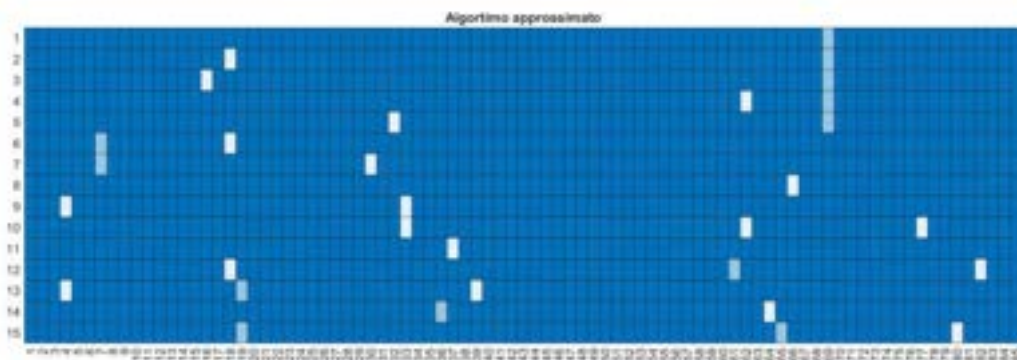
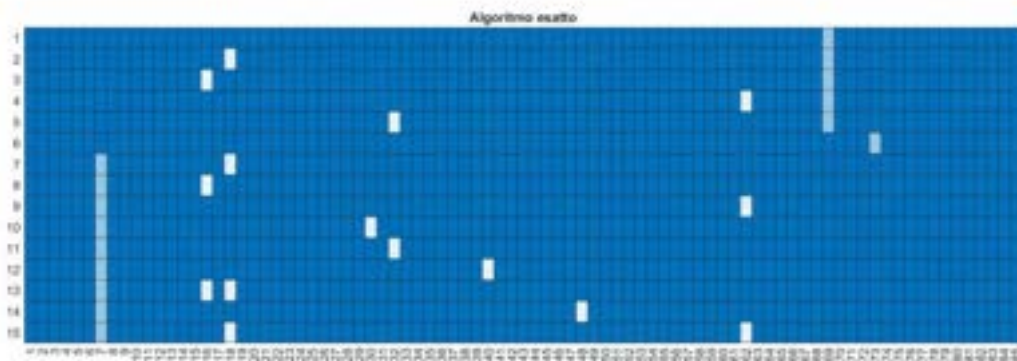
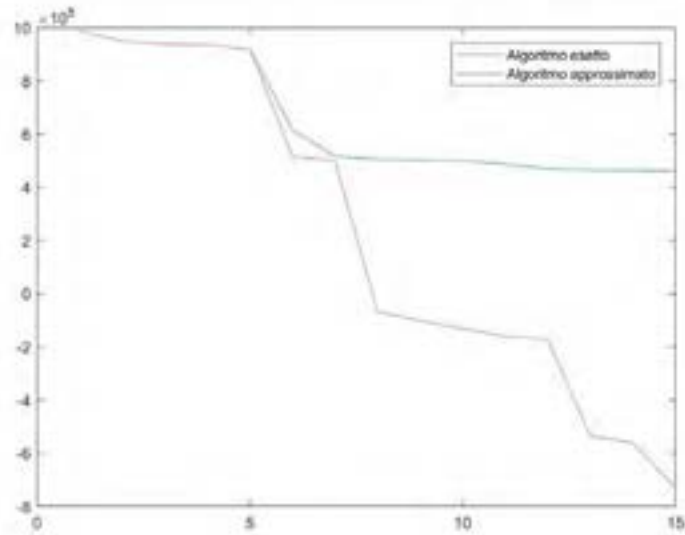


Figura 19 - Soluzioni a confronto tra l'algoritmo esatto e l'algoritmo approssimato. Andamento della FO per le differenti soluzioni e rappresentazione della soluzione attraverso una mappa di calore: Il blu scuro = Diesel, bianco = Tecnologia A, celeste = tecnologia B
 Istanza: [l85_c101_b4000000_t0.9_1](#)

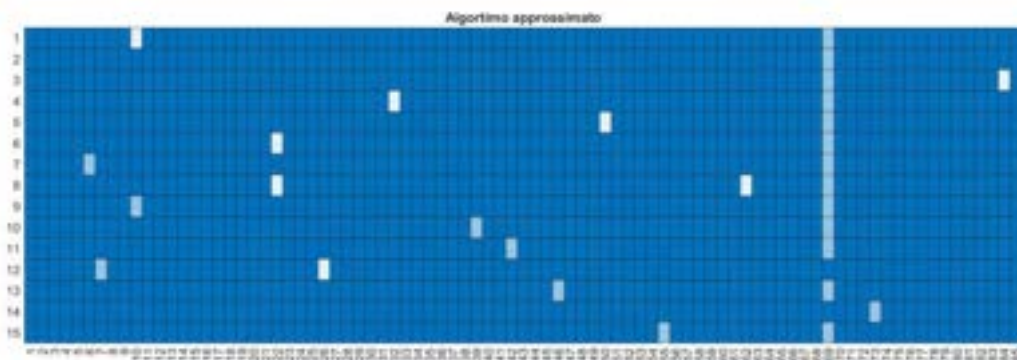
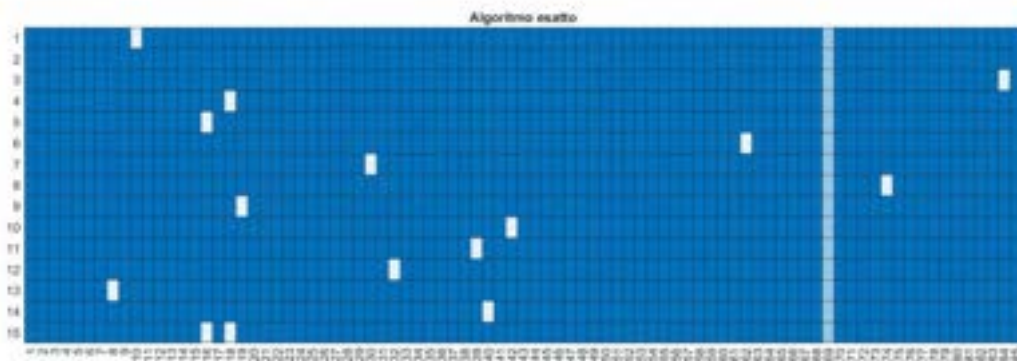
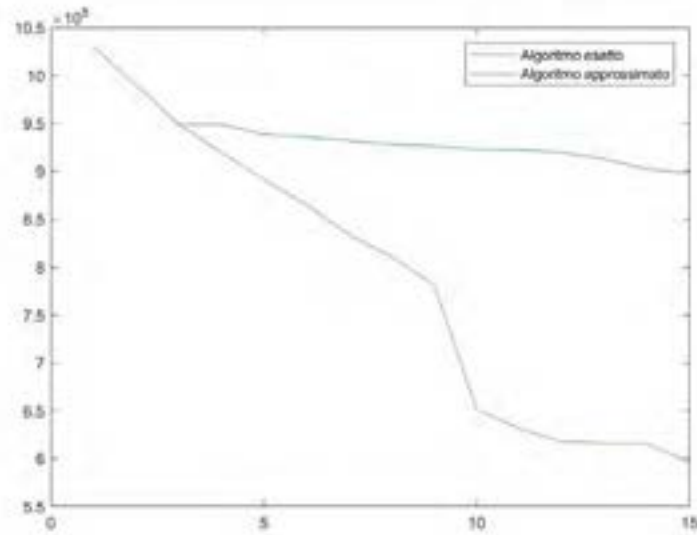


Figura 20 - Soluzioni a confronto tra l'algoritmo esatto e l'algoritmo approssimato. Andamento della FO per le differenti soluzioni e rappresentazione della soluzione attraverso una mappa di calore: Il blu scuro = Diesel, bianco = Tecnologia A, celeste = tecnologia B
 Istanza: [l85_c101_b5000000_t0.1_1](#)

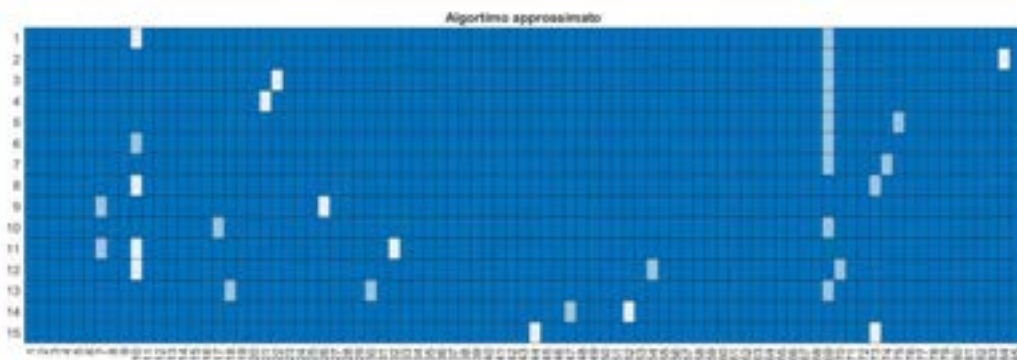
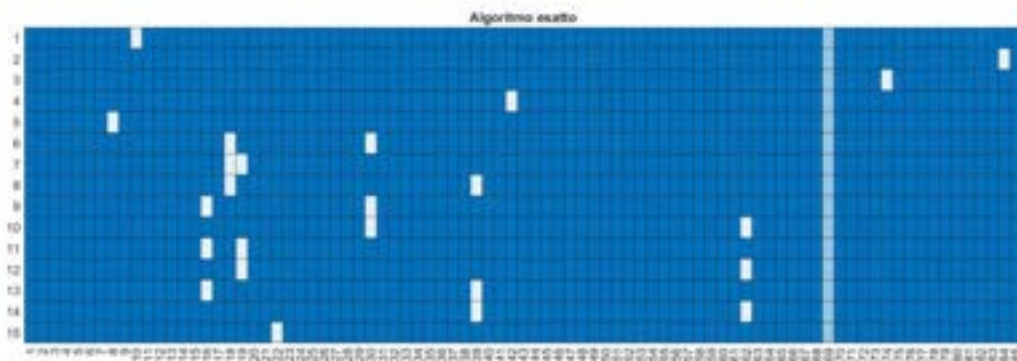
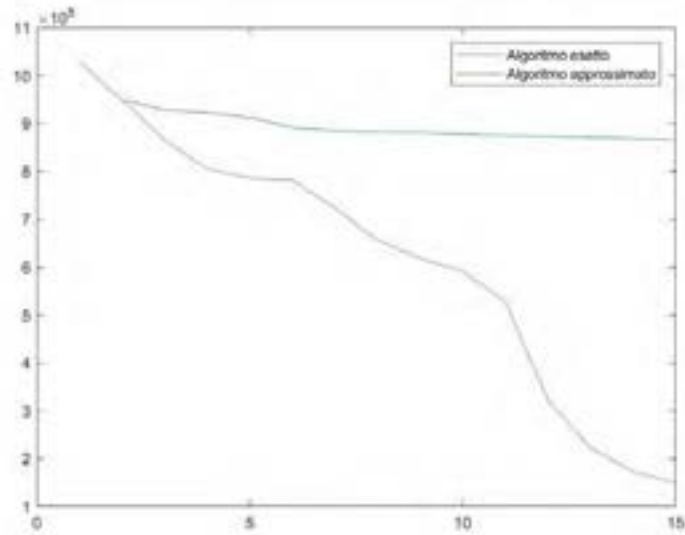


Figura 21 - Soluzioni a confronto tra l'algoritmo esatto e l'algoritmo approssimato. Andamento della FO per le differenti soluzioni e rappresentazione della soluzione attraverso una mappa di calore: Il blu scuro = Diesel, bianco = Tecnologia A, celeste = tecnologia B
 Istanza: [l85_c101_b5000000_t0.9_1](#)

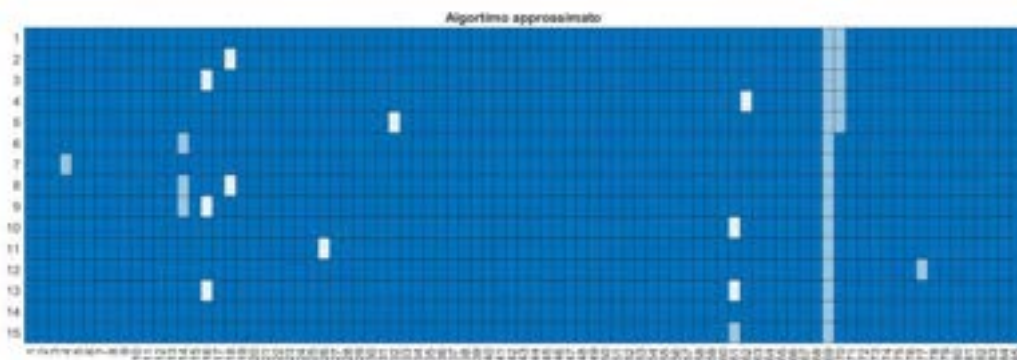
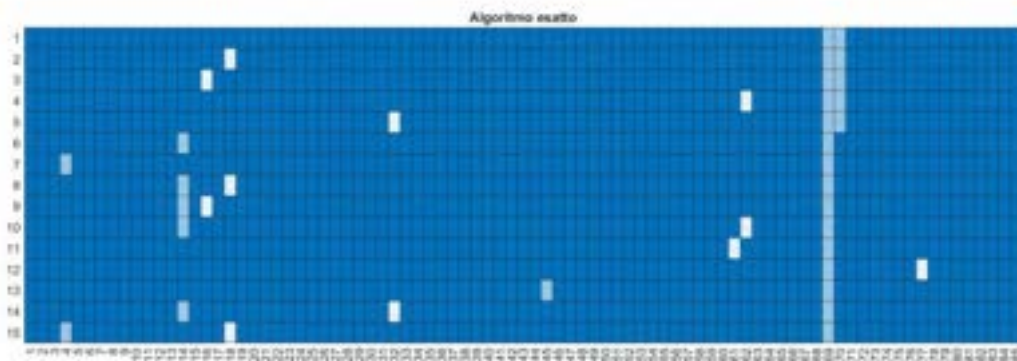
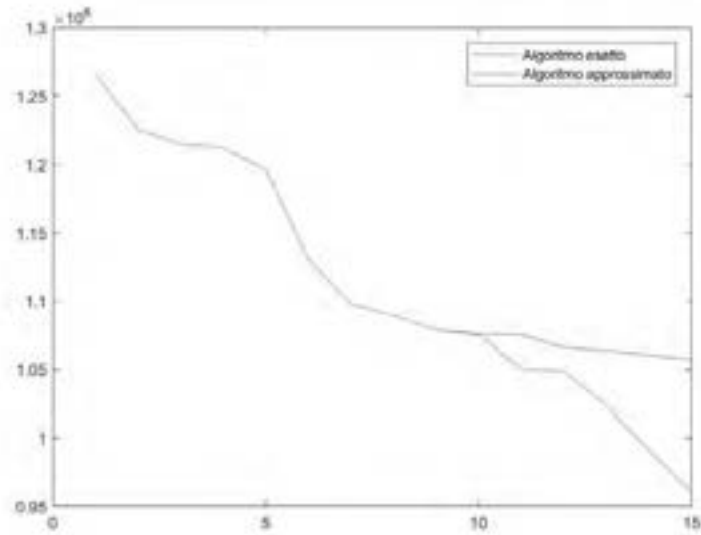


Figura 22 - Soluzioni a confronto tra l'algoritmo esatto e l'algoritmo approssimato. Andamento della FO per le differenti soluzioni e rappresentazione della soluzione attraverso una mappa di calore: Il blu scuro = Diesel, bianco = Tecnologia A, celeste = tecnologia B
 Istanza: [l85_c101_b6000000_t0.1_1](#)

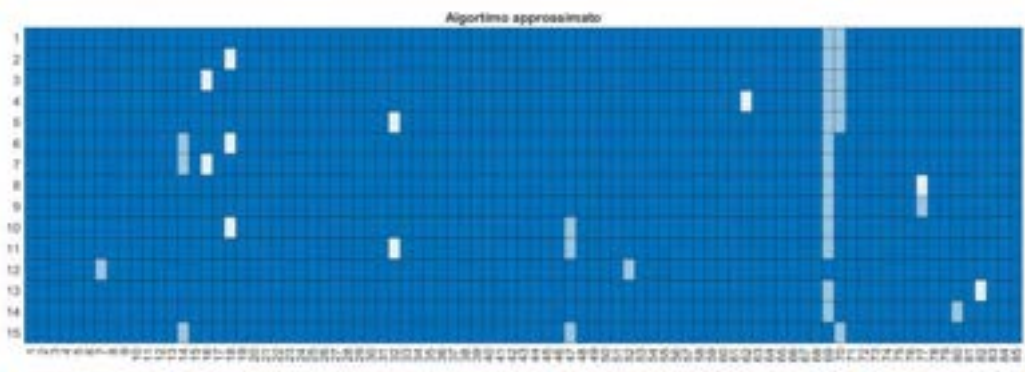
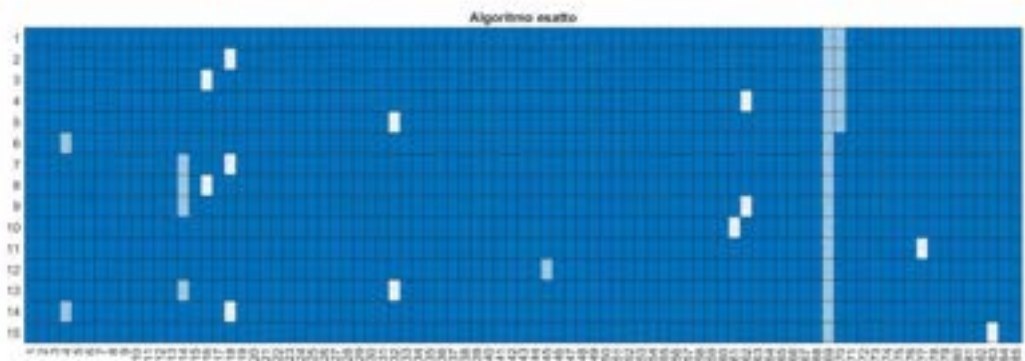
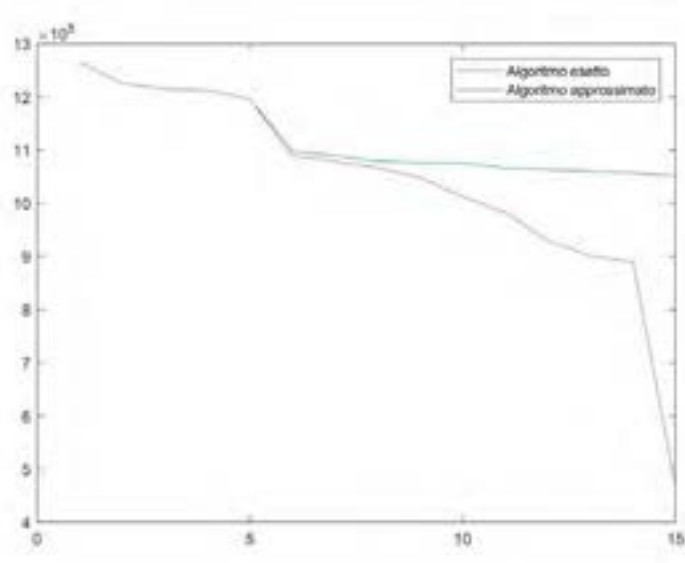


Figura 23 - Soluzioni a confronto tra l'algoritmo esatto e l'algoritmo approssimato. Andamento della FO per le differenti soluzioni e rappresentazione della soluzione attraverso una mappa di calore: Il blu scuro = Diesel, bianco = Tecnologia A, celeste = tecnologia B
 Istanza: [l85_c101_b6000000_t0.9_1](#)

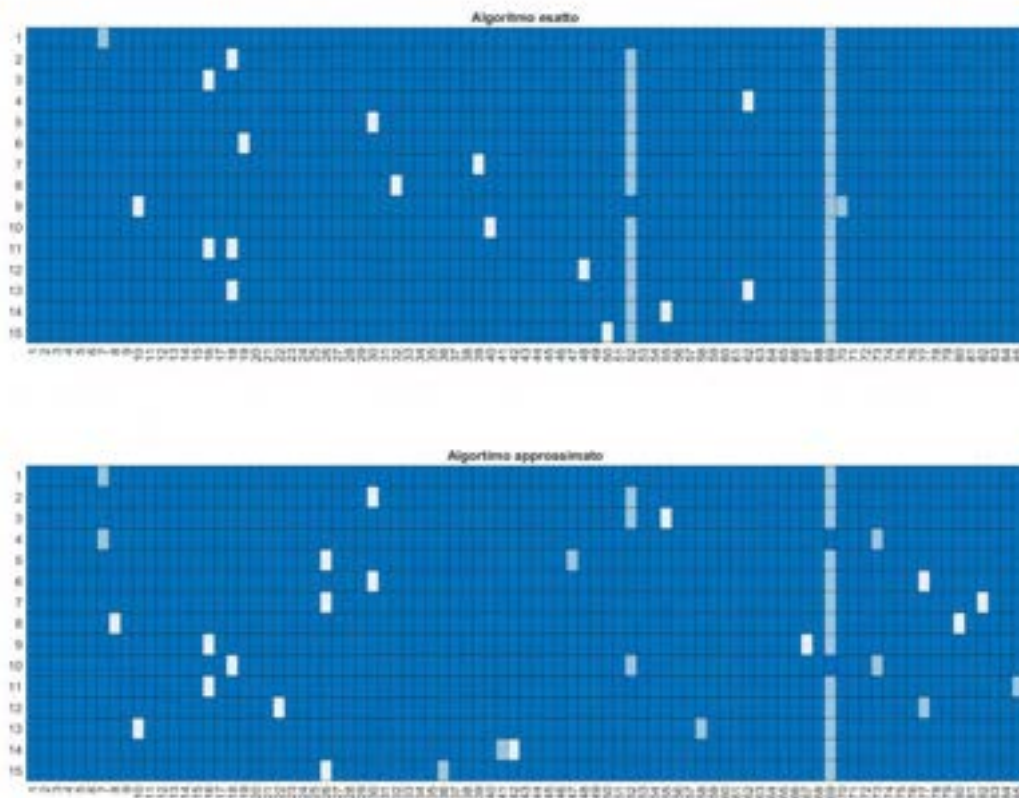
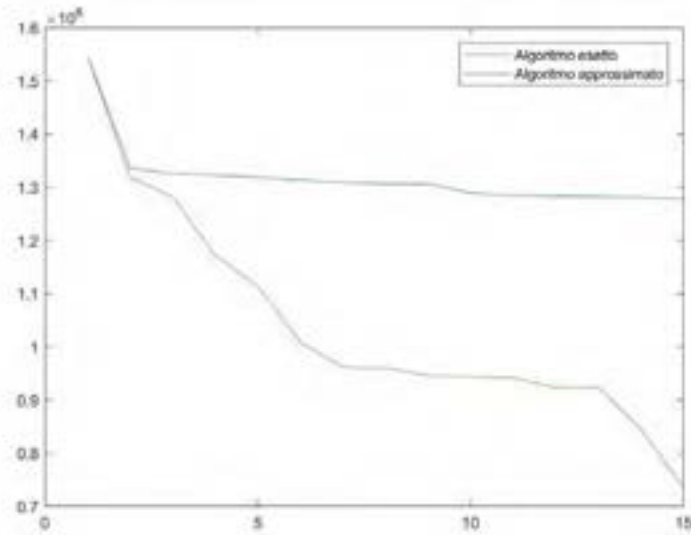


Figura 24 - Soluzioni a confronto tra l'algoritmo esatto e l'algoritmo approssimato. Andamento della FO per le differenti soluzioni e rappresentazione della soluzione attraverso una mappa di calore: Il blu scuro = Diesel, bianco = Tecnologia A, celeste = tecnologia B
 Istanza: [l85_c101_b7000000_t0.9_1](#)

Algoritmo Esatto

Algoritmo Approssimato

l85_c101_b2000000_t0.9

fo	line	arch	line	arch	line	arch	line	arch	fo	line	arch	line	arch	line	Arch	line	arch	line	arch
85953	61	A							85953	61	A								
61929	83	A							19703	18	A	26	A						
19703	18	A	26	A					-84330	41	A								
9133	16	A	26	A					-88954	82	A								
6416	26	A	62	A					-477708	17	B	30	A						
632	26	A	62	B					-578690	6	B	28	B						
-9747	26	A	32	A					-603209	10	B	55	B						
-17799	10	A	30	A					-612908	22	B	30	A						
-23588	10	A	19	A					-698798	18	B	36	B						
-27481	10	A	39	A					-773281	12	B	21	A						
-47595	10	A	40	A					-782840	12	B	44	A	48	A				
-52124	10	A	16	A	18	A			-805661	16	A	18	A	32	A	62	A	79	B
-54841	10	A	18	A	62	A			-811649	22	B	65	B						
-55583	10	A	55	A					-872760	50	B	62	A	65	B				
-60572	2	A	10	A					-903898	43	B	84	B						

l85_c101_b3000000_t0.9

fo	line	arch	line	arch	line	arch	line	arch	fo	line	arch	line	arch	line	arch	line	arch	line	arch
320969	16	A	52	B					320969	16	A	52	B						
318252	52	B	62	A					318252	52	B	62	A						
205655	30	A	70	B					205655	30	A	70	B						
199866	19	A	70	B					7785	67	A								
195974	39	A	70	B					-19698	47	B	74	A						
183661	10	A	14	B					-40820	44	A	45	B						
175860	40	A	70	B					-96732	45	A	50	A						
171331	16	A	18	A	70	B			-108795	42	A	61	B						
168613	18	A	62	A	70	B			-139843	41	B	74	A						
167871	55	A	70	B					-146119	41	A	74	A						
162882	2	A	70	B					-168128	10	A	18	A	39	A	43	A		
162073	44	A	70	B					-205255	8	A	40	A	84	A				
152451	18	A	32	A	70	B			-225620	47	B	74	B						
141881	16	A	32	A	70	B			-254113	22	A	39	A	74	A				
139163	32	A	62	A	70	B			-282111	41	B	75	B						

l85_c101_b4000000_t0.1

fo	line	arch	line	arch	line	arch	line	arch	fo	line	arch	line	arch	line	arch	line	arch	line	arch
989989	69	B							989989	69	B								
949226	18	A	69	B					949226	18	A	69	B						
938656	16	A	69	B					938656	16	A	69	B						
935939	62	A	69	B					935939	62	A	69	B						
919776	32	A	69	B					919776	32	A	69	B						
616235	73	B							557425	7	B								
557425	7	B							506092	7	B	16	A						
516662	7	B	18	A					499653	7	B	30	A						
506092	7	B	16	A					465329	7	B	16	A	18	A				

503375	7	B	62	A					435879	7	B	16	A	32	A
499653	7	B	30	A					372302	52	B				
487212	7	B	32	A					323892	26	A	70	B		
469858	7	B	40	A					263426	70	B				
465329	7	B	16	A	18	A			216055	66	B				
464428	7	B	48	A					212094	16	A	70	B		

I85_c101_b4000000_t0.9

fo	line	arch	line	arch	line	arch	line	arch	fo	line	arch	line	arch	line	arch	line	arch	line	arch
989989	69	B							989989	69	B								
949226	18	A	69	B					949226	18	A	69	B						
938656	16	A	69	B					938656	16	A	69	B						
935939	62	A	69	B					935939	62	A	69	B						
919776	32	A	69	B					919776	32	A	69	B						
616235	73	B							516662	7	B	18	A						
516662	7	B	18	A					499653	7	B	30	A						
506092	7	B	16	A					-67734	66	A								
503375	7	B	62	A					-101463	4	A	33	A						
499653	7	B	30	A					-131796	33	A	62	A	77	A				
487212	7	B	32	A					-159943	37	A								
469858	7	B	40	A					-171213	18	A	61	B	82	A				
465329	7	B	16	A	18	A			-535234	4	A	19	B	39	A				
464428	7	B	48	A					-561035	36	B	64	A						
462612	7	B	18	A	62	A			-729815	19	B	65	B	80	A				

I85_c101_b5000000_t0.1

fo	line	arch	line	arch	line	arch	line	arch	fo	line	arch	line	arch	line	arch	line	arch	line	arch
1029961	10	A	69	B					1029961	10	A	69	B						
989989	69	B							989989	69	B								
949339	69	B	84	A					949339	69	B	84	A						
949226	18	A	69	B					919776	32	A	69	B						
938656	16	A	69	B					891408	50	A	69	B						
935939	62	A	69	B					865117	22	A	69	B						
932218	30	A	69	B					833978	6	B	69	B						
928200	69	B	74	A					811067	22	A	62	A	69	B				
926429	19	A	69	B					781184	10	B	69	B						
922691	42	A	69	B					651278	39	B	69	B						
922536	39	A	69	B					631467	42	B	69	B						
919776	32	A	69	B					617890	7	B	26	A						
912951	8	A	69	B					616444	46	B	69	B						
902422	40	A	69	B					616235	73	B								
897893	16	A	18	A	69	B			595585	55	B	69	B						

I85_c101_b5000000_t0.9

fo	line	arch	line	arch	line	arch	line	arch	fo	line	arch	line	arch	line	arch	line	arch	line	arch
1029961	10	A	69	B					1029961	10	A	69	B						
949339	69	B	84	A					949339	69	B	84	A						
928200	69	B	74	A					865117	22	A	69	B						
922691	42	A	69	B					805198	21	A	69	B						

912951	8	A	69	B						785932	69	B	75	B					
891455	18	A	30	A	69	B				781184	10	B	69	B					
885666	18	A	19	A	69	B				723564	69	B	74	B					
881773	18	A	39	A	69	B				656207	10	A	73	B					
880885	16	A	30	A	69	B				617890	7	B	26	A					
878168	30	A	62	A	69	B				591294	17	B	69	B					
875096	16	A	19	A	69	B				527184	7	B	10	A	32	A			
872379	19	A	62	A	69	B				322751	10	A	54	B	70	B			
871204	16	A	39	A	69	B				222436	18	B	30	B	69	B			
868486	39	A	62	A	69	B				172326	47	B	52	A					
865117	22	A	69	B						149481	44	A	73	A					

l85_c101_b6000000_t0.1

fo	line	arch	line	arch	line	arch	line	arch	fo	line	arch	line	arch	line	arch	line	arch	line	arch
1266391	69	B	70	B					1266391	69	B	70	B						
1225628	18	A	69	B	70	B			1225628	18	A	69	B	70	B				
1215058	16	A	69	B	70	B			1215058	16	A	69	B	70	B				
1212341	62	A	69	B	70	B			1212341	62	A	69	B	70	B				
1196178	32	A	69	B	70	B			1196178	32	A	69	B	70	B				
1130535	14	B	69	B					1130535	14	B	69	B						
1098094	4	B	69	B					1098094	4	B	69	B						
1089772	14	B	18	A	69	B			1089772	14	B	18	A	69	B				
1079202	14	B	16	A	69	B			1079202	14	B	16	A	69	B				
1076485	14	B	62	A	69	B			1075942	61	A	69	B						
1075942	61	A	69	B					1050455	26	A	69	B						
1066583	69	B	77	A					1048752	69	B	77	B						
1063893	45	B	69	B					1024609	16	A	61	A	69	B				
1060322	14	B	32	A	69	B			989989	69	B								
1057331	4	B	18	A	69	B			961310	61	B	69	B						

l85_c101_b6000000_t0.9

fo	line	arch	line	arch	line	arch	line	arch	fo	line	arch	line	arch	line	arch	line	arch	line	arch
1266391	69	B	70	B					1266391	69	B	70	B						
1225628	18	A	69	B	70	B			1225628	18	A	69	B	70	B				
1215058	16	A	69	B	70	B			1215058	16	A	69	B	70	B				
1212341	62	A	69	B	70	B			1212341	62	A	69	B	70	B				
1196178	32	A	69	B	70	B			1196178	32	A	69	B	70	B				
1098094	4	B	69	B					1089772	14	B	18	A	69	B				
1089772	14	B	18	A	69	B			1079202	14	B	16	A	69	B				
1079202	14	B	16	A	69	B			1066583	69	B	77	A						
1076485	14	B	62	A	69	B			1048752	69	B	77	B						
1075942	61	A	69	B					1012386	18	A	47	B	69	B				
1066583	69	B	77	A					982936	32	A	47	B	69	B				
1063893	45	B	69	B					929727	7	B	52	B						
1060322	14	B	32	A	69	B			901035	69	B	82	A						
1057331	4	B	18	A	69	B			889821	69	B	80	B						
1051918	69	B	83	A					467180	14	B	47	B	70	B				

l85_c101_b7000000_t0.9

fo	line	arch	line	arch	line	arch	line	arch	fo	line	arch	line	arch	line	arch	line	arch	line	arch
1547414	7	B	69	B					1547414	7	B	69	B						
1336218	18	A	52	B	69	B			1319209	30	A	52	B	69	B				
1325648	16	A	52	B	69	B			1281426	52	B	55	A	69	B				
1322931	52	B	62	A	69	B			1172115	7	B	73	B						
1319209	30	A	52	B	69	B			1113615	26	A	47	B	69	B				
1313420	19	A	52	B	69	B			1008812	30	A	69	B	77	A				
1309528	39	A	52	B	69	B			961501	26	A	69	B	82	A				
1306768	32	A	52	B	69	B			960306	8	A	69	B	80	A				
1306363	10	A	69	B	70	B			946441	16	A	67	A	69	B				
1289414	40	A	52	B	69	B			944295	18	A	52	B	73	B				
1284885	16	A	18	A	52	B	69	B	941886	16	A	69	B	85	B				
1283985	48	A	52	B	69	B			923881	22	A	69	B	77	B				
1282168	18	A	52	B	62	A	69	B	923661	10	A	58	B	69	B				
1281426	52	B	55	A	69	B			844637	41	B	42	A	69	B				
1278400	50	A	52	B	69	B			735963	26	A	36	B	69	B				

2.5 Il software sviluppato

L'algoritmo approssimato è stato implementato con il linguaggio di programmazione C# usando il framework Microsoft .NET 4.5. La scelta è stata determinata dalla ricerca del giusto equilibrio tra l'efficienza del linguaggio e la necessità di creare un'interfaccia user friendly e di integrarsi successivamente con altri software in ambiente Microsoft Windows.

Per garantire l'integrazione il software di calcolo è stato sviluppato internamente ad una libreria DLL e pertanto incorporabile in qualsiasi software per Windows. Parallelamente sono stati sviluppati 2 software:

- ONEBus: software standalone dotato di GUI che permette le seguenti funzioni:
 - caricamento di istanze da memoria fissa;
 - gestione dei parametri di esecuzione dell'algoritmo di ottimizzazione;
 - esecuzione dell'algoritmo di ottimizzazione;
 - navigazione del database delle soluzioni trovate;
 - visualizzazione ed esportazione delle specifiche soluzioni;
- ONEBusBatch: è uno specifico per l'esecuzione batch di istanze multiple caratterizzato da una GUI basilare che permette la selezione delle istanze e la relativa esecuzione.

2.5.1 ONEBus

ONEBus presenta una GUI molto intuitiva che permette il caricamento di uno scenario e l'impostazione dei parametri governabili dall'utente:

- Investimento massimo o Budget;
- Tolleranza;
- Ora di punta: valore usato per la valutazione la valutazione del parametro K nella funzione obiettivo;
- N° di soluzioni da trovare;
- N° di TS da eseguire parallelamente;
- N° di soluzioni totale da esplorare;

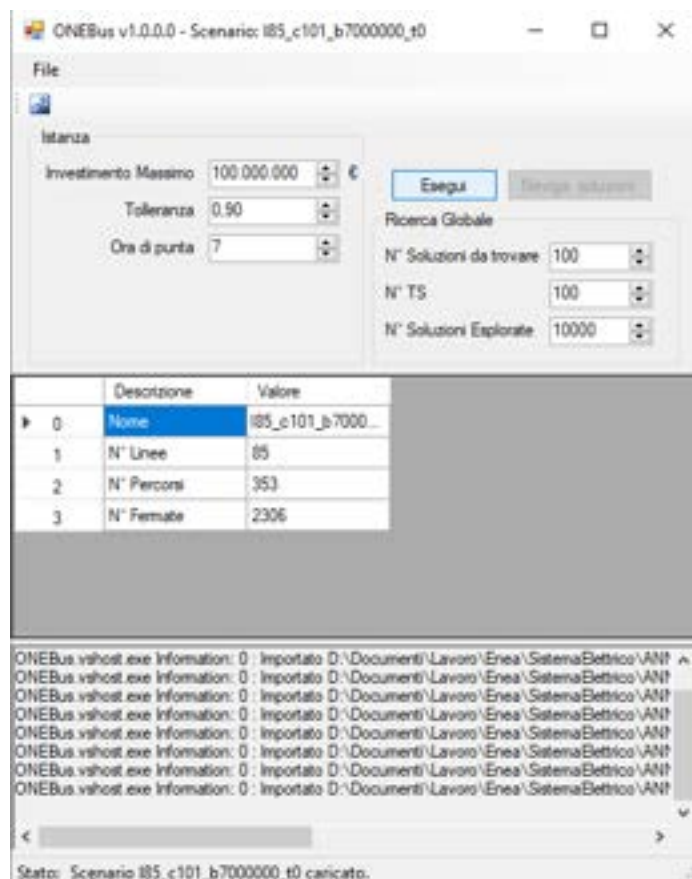


Figura 25 - Finestra Principale di ONEBus

Lo scenario da caricare è caratterizzato da una cartella con i seguenti file:

- arch_a.csv
- arch_b.csv
- arch_c.csv
- dati_generali.csv
- impianti_standard.csv
- linee.csv
- nodi.csv
- linee_output_best.csv
- percorsi.csv
- percorsi_fermate.csv

Per la descrizione del contenuto dei primi sei file si rimanda al Report RdS/PAR2016/226 di ENEA relativo all'aggiornamento del software BEST; per la descrizione dei restanti file si rimanda al paragrafo 2.6 del presente Report.

Qui si ricorda solamente che alcuni di tali dati sono ottenuti dal pre-processing dei dati in formato GTFS acquisiti via web, altri sono inputati direttamente dall'utente, altri ancora, infine, sono generati dal modulo BEST di calcolo sviluppato dall'ENEA.

Premendo sul pulsante "Esegui" il software avvia l'algoritmo di ottimizzazione e al termine dell'esecuzione permette di navigare le soluzioni attraverso le seguenti schermate:

Id	A	B	C	Diesel	Tot	FO	Budget
1	0	2	0	83	85	1.547.413,66	6.322.926 €
2	1	2	0	82	85	1.319.209,36	6.968.895 €
3	0	2	0	83	85	1.172.115,44	6.984.437 €
4	1	2	0	82	85	871.955,20	6.663.755 €
5	2	2	0	81	85	621.634,68	6.982.500 €
6	2	2	0	81	85	471.733,82	6.507.279 €
7	2	0	0	83	85	276.145,23	6.829.832 €
8	2	1	0	82	85	235.153,81	6.630.849 €
9	5	1	0	79	85	63.527,27	6.872.991 €
10	4	0	0	81	85	58.030,15	6.999.674 €
11	3	1	0	81	85	-287.972,46	6.765.458 €
12	2	2	0	81	85	-345.648,97	6.774.869 €
13	3	1	0	81	85	-404.266,39	6.756.824 €
14	3	1	0	81	85	-427.016,14	6.521.689 €
15	4	1	0	80	85	-469.215,90	6.829.319 €
16	3	1	0	81	85	-485.272,67	6.674.397 €
17	2	2	0	81	85	-563.560,32	6.355.824 €
18	2	2	0	81	85	-645.544,12	6.760.298 €
19	1	1	0	83	85	-905.487,52	7.005.012 €
20	2	2	0	81	85	-912.300,88	6.547.461 €
21	2	2	0	81	85	-987.489,73	6.740.451 €
22	4	1	0	80	85	-1.046.898,...	6.953.901 €
23	1	3	0	81	85	-1.085.369,...	6.886.543 €
24	2	4	0	79	85	-1.380.795,...	6.348.186 €

nLinea	Architetture	PO	Budget
0	Diesel	0	0 €
1	A	-100.545	676.469 €
2	Diesel	0	0 €
3	Diesel	0	0 €
4	Diesel	0	0 €
5	Diesel	0	0 €
6	Diesel	0	0 €
7	Diesel	0	0 €
8	Diesel	0	0 €
9	Diesel	0	0 €
10	A	26.972	1.128.953 €
11	Diesel	0	0 €
12	Diesel	0	0 €
13	Diesel	0	0 €
14	Diesel	0	0 €
15	Diesel	0	0 €
16	Diesel	0	0 €
17	Diesel	0	0 €
18	Diesel	0	0 €
19	Diesel	0	0 €
20	Diesel	0	0 €
21	Diesel	0	0 €
22	Diesel	0	0 €
23	Diesel	0	0 €
24	Diesel	0	0 €
25	Diesel	0	0 €
26	Diesel	0	0 €
27	Diesel	0	0 €
28	Diesel	0	0 €
29	Diesel	0	0 €
30	A	57.771	894.862 €

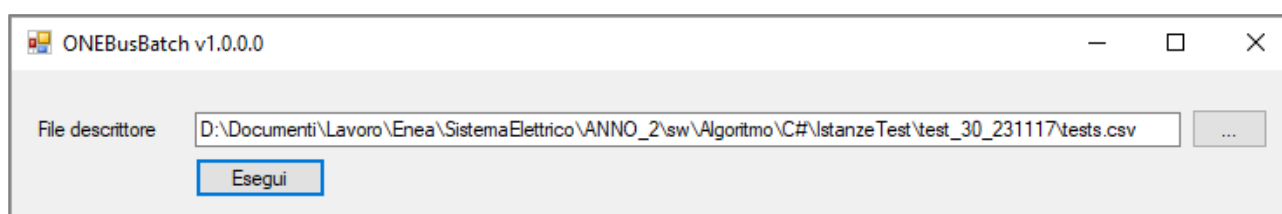
Figura 26 - Esempi di schermate di navigazione delle soluzioni

Attraverso il menu file è anche possibile esportare i risultati in formato CSV.

2.5.2 ONEBusBatch

ONEBusBatch permette l'esecuzione di una sequenza di istanze descritte all'interno del file descrittore. L'interfaccia permette di selezionare il file descrittore e di eseguire l'algoritmo di risoluzione. All'interno del file descrittore, in formato CSV, esistono i seguenti campi necessari all'esecuzione dell'algoritmo:

- n_linee: numero di linee;
- n_capolinea: numero di capolinea;
- budget: budget massimo;
- tolleranza: rapporto tra budget minimo e massimo;
- test: identificativo del test;
- folder: cartella dello scenario;



I risultati di ogni singola istanza verranno memorizzati nella cartella dello scenario con il nome "risultati.csv"

2.5.3 Integrazione con i sistemi ENEA

I software realizzati, anche se sviluppati come applicativi stand-alone, sono stati integrati con i sistemi informativi preesistenti in ENEA. In particolare nel corso del 2016 è stato sviluppato, dall'Università dell'Aquila, un database relazionale e un insieme di procedure di supporto all'analisi tecnica ed economica di scenari di elettrificazione delle linee di trasporto pubblico locale (TPL) su gomma.

In particolare l'Università dell'Aquila ha realizzato una procedura per il caricamento automatico della rete del trasporto pubblico a partire dal formato GTFS. Questa informazione è stata successivamente integrata all'interno di un geo-database per essere fruibile dai diversi sistemi che si sarebbero dovuti interfacciare per l'analisi degli scenari di elettrificazione del TPL. Il database è stato anche progettato per essere uno strumento di interfaccia tra i diversi software per lo scambio di dati ed è stato utilizzato per integrare il software BEST con i modelli di calcolo delle emissioni e dei consumi e per poter visualizzare i risultati su piattaforma GIS.

Vista la natura di strumento di integrazione del database ma anche la necessità di creare degli applicativi stand-alone, si è deciso di integrare, all'interno del software sviluppato dall'Università dell'Aquila, delle procedure di pre-processing utili a creare i file dei dati di input di ONEBus.

Il pre-processing ha riguardato principalmente 2 task:

1. Elaborazione spaziale della rete del TPL per la clusterizzazione dei capolinea;
2. Estrazione e trasformazione dei dati del TPL in funzione delle specifiche concordate per il progetto ONEBus, sia per l'algoritmo esatto che per quello approssimato;

Il GTFS solitamente fornito dalla pubblica amministrazione ha un dettaglio di percorso che identifica le fermate iniziali e finali con le singole paline e che quindi, per i capolinea caratterizzati da più paline, i percorsi non risultavano convergenti un'unica stazione. La prima attività ha riguardato lo sviluppo di una procedura PL/SQL per PostgreSQL 9.5 che, analizzando la localizzazione delle paline iniziali e finali, ha creato dei cluster spaziali identificando i capolinea caratterizzati da più paline. La soglia massima è stata impostata a 500 metri, aggregando tutte le paline non più lontani di tale distanza ad un unico capolinea. Il capolinea, così identificato, è stato inserito nel database come un nuovo nodo di tipo "CAPOLINEA".

Identificati questa nuova tipologia di nodo tutti le fermate iniziali e finali dei percorsi sono stati rimappati con i nuovi identificativi. Questa procedura è stata infine integrata direttamente nel software di importazione del GTFS. Il nome della procedura realizzata è *fn.get_domains_n()* e il codice è stato integrato all'interno del file *04_elabora_gtfs.sql* presente all'interno del codice dell'Università dell'Aquila.



Figura 27 - Esempio di clusterizzazione di un capolinea

Il database, inoltre, è stato utilizzato per memorizzare i dettagli del TPL, per ogni singola corsa. Nella seconda attività di quelle di pre-processing questi dati sono stati aggregati in funzione delle esigenze di ONEBus sia per percorso che per linea, calcolando indicatori di consumo o prestazione. Durante le procedure di pre-processing sono state introdotte anche delle formulazioni lineari di stima dei consumi chilometrici da utilizzarsi solo come valori di default, qualora non sia stato definito una tipologia di veicolo per ogni singola corsa.

Sia le formulazioni che le operazioni di aggregazione sono state mascherate in query preconfezionate presenti all'interno del DB. Tale query sono dette viste e sono state nominate in funzione dei relativi file di input. Di seguito vengono elencate con la descrizione della funzione principale di preprocessing creata nella vista:

- v_fermate_percorsi: estrazione della matrice di incidenza fermate percorsi, con calcolo dei consumi di default e aggregazione per linea, percorso e fascia oraria;
- v_linee: estrazione del veicolo prevalente usato sulla linea tra tutte le corse;
- v_linee_output_best: merge dei costi dati dall'analisi delle tecnologie elettriche e tradizionali. Durante il merge i costi sono stati aggregati per distinguere i costi operativi da quelli di investimento;
- v_nodi: merge delle fermate con i capolinea riportanti le caratteristiche essenziali di nodo;
- v_numero_stalli: merge delle fermate e capolinea con le caratteristiche dipendenti dalla tecnologia applicata;
- v_percorsi: descrizione dei percorsi con relativa aggregazione dei consumi e stima dei tempi di ricarica per fascia oraria per ciascun percorso;

2.6 Descrizione dei principali dati di input del SW sviluppato generati da BEST

Per le esigenze delle attività degli anni successivi, il SW sviluppato legge tutti i dati messi a disposizione. Tuttavia solo alcuni di questi vengono effettivamente utilizzati dall’algoritmo messo a punto nel primo anno di attività. In particolare, dal file Percorsi.csv vengono utilizzate le seguenti grandezze descrittive della rete e dei consumi dei mezzi, descritte con maggior dettaglio nel Report RdS/PAR2016/226 di ENEA (sezione 4.3).

NOME VARIABILE	DESCRIZIONE	UNITA' DI MISURA
percorso_id	Identificativo percorso	
linea_id	Identificativo linea	
Direzione	Identificativo direzione	
capolinea_i	Identificativo nodo capolinea iniziale	
capolinea_f	Identificativo nodo capolinea finale	
Lungh	Lunghezza del percorso	m
ora_arrivo	Ora della fascia oraria di arrivo nel capolinea finale	Ora orologio
frequenza	n. di corse in arrivo nel nodo finale nella fascia oraria	n. corse
consumi	Consumo complessivo delle corse in arrivo nel nodo finale nella fascia oraria, calcolati da ECOTRIP sul veicolo standard	kWh
tempo_ricarica	Tempo complessivo necessario per la ricarica nel nodo di arrivo nella fascia oraria	Ore

Dal file Linee_output_best.csv generato da BEST vengono lette le seguenti grandezze, necessarie per il calcolo della funzione obiettivo del problema di ottimizzazione, anche queste descritte con maggior dettaglio nel Report RdS/PAR2016/226 di ENEA (sezione 4.3).

NOME VARIABILE	DESCRIZIONE	UNITA' DI MISURA
Linea_id	Identificativo linea	
Arch_id	Identificativo architettura	
fatt_arch	Fattibilità architettura	
Flotta	n. di veicoli necessari al servizio	
n_cap_ricarica	Numero di capolinea equipaggiati con ricarica per arch. B e C	
n_ferm_ricarica	Numero di fermate equipaggiate con ricarica	
invest_iniziale_bus	Investimento iniziale per acquisto flotta	€
costo_energia	Valore attualizzato dei costi sostenuti per l'acquisto di energia elettrica nel periodo di analisi	€
manut_bus	Valore attualizzato dei costi sostenuti per mantenere la flotta nel periodo di analisi, comprese eventuali sostituzioni degli accumuli di bordo e degli stessi veicoli	€
invest_impianti_dep	Investimento iniziale per acquisto e installazione degli impianti di ricarica necessari per la linea, considerata singolarmente	€
invest_impianti_cap	Investimento iniziale per acquisto e installazione degli impianti di ricarica necessari per la linea, considerata singolarmente	€
invest_impianti_ferm	Investimento iniziale per acquisto e installazione degli impianti di ricarica necessari per la linea, considerata singolarmente	€
manut_impianti_dep	Valore attualizzato dei costi per manutenzione degli impianti di ricarica e di eventuali valori residui a fine periodo	€
manut_impianti_cap	Valore attualizzato dei costi per manutenzione degli impianti di ricarica e di eventuali valori residui a fine periodo	€
manut_impianti_ferm	Valore attualizzato dei costi per manutenzione degli impianti di ricarica e di eventuali valori residui a fine periodo	€
costo_allacci_dep	Valore attualizzato dei costi sostenuti per gli allacci alla rete nel periodo di analisi	€
costo_allacci_cap	Valore attualizzato dei costi sostenuti per gli allacci alla rete nel periodo di analisi	€
costo_allacci_ferm	Valore attualizzato dei costi sostenuti per gli allacci alla rete nel periodo di analisi	€
costo_potenza_dep	Valore attualizzato dei costi sostenuti per l'impegno di potenza nel periodo di analisi	€
costo_potenza_cap	Valore attualizzato dei costi sostenuti per l'impegno di potenza nel periodo di analisi	€
costo_potenza_ferm	Valore attualizzato dei costi sostenuti per l'impegno di potenza nel periodo di analisi	€

3 Conclusioni

Le attività di ricerca di base condotte dal DI-RM3, in collaborazione con ENEA, nell'ambito dell'accordo di collaborazione descritto a pag. 2 del presente documento, hanno riguardato lo sviluppo di uno strumento di supporto alle decisioni operative delle pubbliche amministrazioni interessate a convertire quote di trasporto pubblico dagli attuali motori diesel alle nuove tecnologie elettriche. A fronte di un budget per l'investimento iniziale in elettro-mobilità il problema affrontato è quello di definire su quali linee e quale tecnologia sia più appropriata per ogni linea al fine di massimizzare il risparmio ottenibile tenendo conto anche dell'effetto rete ottenibile. Il cuore del sistema di supporto alle decisioni è costituito da modelli e algoritmi di ottimizzazione rapidi ed efficienti per la soluzione di questo problema. In quest'anno di attività la ricerca si è concentrata sulle architetture A e B, con l'obiettivo di definire alcuni algoritmi euristici veloci in grado di risolvere il problema per una media città italiana (Firenze) in tempi contenuti.

La ricerca ha prodotto diversi modelli e diversi algoritmi euristici di ottimizzazione, sia greedy sia metaeuristici, in particolare questi ultimi basati sugli schemi di Tabu Search e Iterated Local Search. Si è poi passati allo sviluppo di un dimostratore degli algoritmi e all'esecuzione di test funzionali e prestazionali

sulla base delle istanze fornite da ENEA. L'algoritmo complessivamente più performante tra quelli sviluppati è risultato essere quello denominato KTS. Come algoritmo di confronto per la valutazione complessiva di KTS è stato utilizzato l'algoritmo esatto sviluppato dall'Università di Roma Tor Vergata, basato su calcolo parallelo, che enumera implicitamente tutte le soluzioni del problema.

L'algoritmo approssimato KTS ha mostrato ottimi risultati nella ricerca della soluzione ottima, trovandola in tutte le istanze tranne che in una. Ottimi risultati sono stati trovati anche nell'esplorazione dello spazio delle soluzioni per la ricerca di un insieme ammissibile di soluzioni buone.

L'algoritmo ha lavorato egregiamente sia sottoposto a vincoli stringenti (budget e tolleranza) che su istanze reali come la rete di Firenze. Il numero di soluzioni esplorate dal TS è stato sempre limitato a 10.000 mantenendo i tempi di esecuzione limitati a pochi minuti e dipendenti solo dalla dimensione dell'istanza.

Durante l'analisi effettuata si è potuto appurare che un buon incremento dell'efficacia dell'algoritmo si è ottenuto attraverso l'utilizzo del Knapsack e delle sue varianti per il calcolo delle soluzioni iniziali.

Gli algoritmi TS contribuiscono notevolmente all'esplorazione dello spazio delle soluzioni migliorando la FO media del set di soluzioni migliori.

4 Riferimenti bibliografici

1. European Commission, White paper, Roadmap to a Single European transport Area – Towards a competitive and resource efficient transport system, COM(2011) 144 final, Brussels, 28.3.2011.
2. G. Brusaglino, R. Gava, M. Leon, F. Porcel, "TECMEHV–Training & Development of European Competences on Maintenance of Electric and Hybrid Vehicles", International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium Barcelona, Spain, November 17-20, 2013.
3. A. Heidi, R. Sampsa, O. Juha, T. Anu, A. Toni "Process to support strategic decision-making: Transition to electromobility", International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium Barcelona, Spain, November 17-20, 2013.
4. T. Coosemans, I. Keseru, C. Macharis, J. Van Mierlo, B. Muller G. Meyer, "Societal Drivers for the Future Transport System in Europe: the Mobility4EU Project", International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium Montréal, Québec, Canada, June 19-22, 2016.
5. T. Soyly, J.E. Anderson, N. Böttcher, C. Weiß, B. Chlonda, T. Kuhnimhof, "Building Up Demand-Oriented Charging Infrastructure for Electric Vehicles in Germany", Transportation Research Procedia, 19 (2016), pp. 187-198, Elsevier.
6. A. Talantsev, "Who gains and who loses in the shift to electric vehicles: impact assessment through multicriteria multi-stakeholder analysis", Procedia Environmental Sciences, 37 (2017), pp. 257-268, Elsevier.
7. S. Carley, R.M. Krause, B.W. Lane, J.D. Graham, "Intent to purchase a plug-in electric vehicle: A survey of early impressions in large US cities", Transportation Research Part D: Transport and Environment, 18 (2013), pp. 39-45. Elsevier.
8. H. Gong, M. Wang, and H. Wang, "New energy vehicles in China: policies, demonstration, and progress", Mitigation and Adaptation Strategies for Global Change, 18 (2013), pp. 207-228.
9. K. Yabe, Y. Shinoda, T. Seki, H. Tanaka, and A. Akisawa, "Market penetration speed and effects on CO2 reduction of electric vehicles and plug-in hybrid electric vehicles in Japan". Energy Policy, 45 (2012), pp. 529-540.
10. G. Crouch, "Volvo to STOP making petrol and diesel cars from 2019 in favour of electric or hybrid models", The Sun (6th July 2017), Available from: www.thesun.co.uk/motors/3957485/volvo-announces-it-will-stop-making-petrol-and-diesel-cars/
11. A. Chrisafis and A. Vaughan, "France to ban sales of petrol and diesel cars by 2040", The Guardian (6th July 2017), Available from: www.theguardian.com/business/2017/jul/06/france-ban-petrol-diesel-cars-2040-emmanuel-macron-volvo

12. J. Staufenberg, "Norway to completely ban petrol powered cars by 2025", The Independent (4th June 2016), Available from: <http://www.independent.co.uk/environment/climate-change/norway-to-ban-the-sale-of-all-fossil-fuel-based-cars-by-2025-and-replace-with-electric-vehicles-a7065616.html>
13. Bloomberg, "Electric Vehicle Outlook 2017", (2017). Bloomberg Finance L.P. 2017.
14. P. Toth, S. Martello, "Algorithms for Knapsack Problems", In Silvano Martello, Gilbert Laporte, Michel Minoux, Celso Ribeiro (editors), "Surveys in Combinatorial Optimization", North-Holland Mathematics Studies, North-Holland, Volume 132, (1987), pp. 213-257, ISSN 0304-0208, ISBN 9780444701367.
15. B. Codenotti, G. Manzini, L. Margara, G. Resta, "Perturbation: An efficient technique for the solution of very large instances of the euclidean TSP", INFORMS Journal on Computing, 8 (1996), pp. 125-133.
16. T. Stützle, "Iterated local search for the quadratic assignment problem", European Journal of Operational Research, 174 (2006), pp. 1519-1539.
17. Fred Glover, "Future Paths for Integer Programming and Links to Artificial Intelligence", Computers and Operations Research, 13 (1986), pp. 533-549.

5 Abbreviazioni ed acronimi

ENEA	Agenzia nazionale per le nuove tecnologie, l'energia e lo sviluppo economico sostenibile
DI-RM3	Dipartimento di Ingegneria dell'Università degli Studi Roma Tre
PLI	Programmazione Lineare a numeri Interi
ILS	Iterated Local Search
IKP	Integer Knapsack Problem
FO	Funzione Obiettivo
KP	Knapsack Problem
KTS	Nome dell'algoritmo usato
TS	Tabu Search
UB	Upper Bound
LB	Lower Bound

6 Appendice 1. Breve curriculum scientifico del gruppo di lavoro

6.1 Prof. Dario Pacciarelli

Dario Pacciarelli è professore ordinario di *Ricerca Operativa* presso l'Università degli Studi Roma Tre, nella quale è entrato come ricercatore nel 1996. Presso questo ateneo, dal 2010 è responsabile del s.s.d. Mat/09 Ricerca Operativa e dirige il *Laboratorio di Automazione e Organizzazione Industriale*, istituito presso il *Dipartimento di Ingegneria*. Dal 2016 è Coordinatore del Collegio Didattico di Ingegneria Informatica.

Didattica

Presso l'università Roma Tre è titolare degli insegnamenti *Ricerca Operativa I (Laurea in Ingegneria Informatica)* dal 1999; *Ottimizzazione della Logistica (Laurea Magistrale in Ingegneria Gestionale e dell'Automazione)* dal 2004; *Ricerca Operativa (Laurea Magistrale in Ingegneria delle Infrastrutture Viarie e Trasporti)* dal 2004.

Ricerca

E' autore o co-autore di oltre cento pubblicazioni, circa metà delle quali su riviste internazionali indicizzate con valutazione. Gli interessi di ricerca riguardano lo sviluppo di soluzioni informatiche per il supporto alle decisioni manageriali, principalmente basati su metodi di ottimizzazione. Ha lavorato alla riconfigurazione di sistemi di produzione, pianificazione della produzione industriale, nuovi modelli per lo scheduling, ottimizzazione e coordinamento dei sistemi di trasporto, supply chain scheduling. I contributi teorici di ricerca vertono sul progetto di modelli e algoritmi per la soluzione di problemi di ottimizzazione combinatoria, e sull'analisi della complessità computazionale di algoritmi e problemi. I settori applicativi riguardano la produzione industriale, la logistica e i trasporti pubblici. In particolare, l'ottimizzazione dei trasporti pubblici è stato il principale dominio applicativo studiato negli ultimi 15 anni. In occasione della procedura di valutazione della qualità della ricerca VQR 2011-2014, il gruppo di Ricerca Operativa coordinato dal Prof. Pacciarelli è risultato primo in graduatoria in Italia nel s.s.d. Mat/09 Ricerca Operativa.

Coordinamento di progetti

Dal 1999 ad oggi è stato responsabile scientifico per l'Università Roma Tre e/o coordinatore nazionale di progetti di ricerca e/o di trasferimento tecnologico, prevalentemente sui temi dello sviluppo di soluzioni informatiche per l'automazione o per il supporto alle decisioni, finanziati dalla Commissione Europea, dal M.I.U.R., dal Mi.S.E., dal C.N.R. e da grandi gruppi industriali (DaimlerChrysler, Bombardier Transportation, Alstom Ferroviaria).

Comitati scientifici

E' stato membro di diversi comitati di programma di congressi internazionali ed è membro permanente del board della *International Association of Railway Operations Research* e dell'International Scientific Committee della *Conference on Advanced Systems for Public Transport*. E' membro del Consiglio Direttivo dell'Associazione Italiana di Ricerca Operativa.

6.2 Phd. Ing. Andrea Gemma

Conduce attività di ricerca presso l'Università di Roma Tre ed esercita, contemporaneamente, l'attività di Ingegnere Informatico collaborando con diverse società private e pubbliche. Il know-how conseguito durante le esperienze lavorative si è incentrato principalmente nell'ambito dell'ingegneria dei sistemi di trasporto, dell'infomobilità, degli ITS (Intelligent Transport Transportation Systems), dei sistemi GIS, applicazioni sui BIG DATA e dell'algoritmica in generale.

Ha conseguito il dottorato in Ingegneria Informatica e dell'Automazione, nell'ambito della "Ricerca Operativa" applicata ai trasporti presso l'Università di "Roma Tre" sullo sviluppo di algoritmi euristici di ottimizzazione della sincronizzazione semaforica.

Attività svolte

2017 - Assegno di ricerca: Valutazione del livello di servizio del trasporto pubblico per mezzo dell'analisi dei Big-Data provenienti dal tracciamento GPS dei veicoli. (DR 148/2016).

2016 - Assegno di ricerca: Ingegneria dei trasporti per la mobilità sostenibile nel settore dei beni culturali (DR 85/2015).

2013 - Assegno di ricerca: Sviluppo di algoritmi per il tracciamento dei veicoli sonda, per l'analisi delle reti stradali e per la valutazione dei tempi di sosta DR 1708/2013.

2012 - Assegno di ricerca: Modelli per stima delle emissioni inquinanti nelle reti di trasporto congestionate DR 908/2012.

2011 - Assegno di ricerca: Progettazione e sviluppo di algoritmi di simulazione dinamica del traffico. Capacità di ricerca nel settore.

Dal 2007 Ingegnere Libero professionista iscritto all'albo con decine di attività nell'ambito dell'ingegneria informatica applicata agli strumenti ITS, GIS e analisi dati per l'ingegneria del traffico e dei trasporti.

Formazione

2011 – Dottorato di ricerca in Ingegneria Informatica e dell'Automazione

2006 – Laurea V.O. in Ingegneria Informatica e dell'Automazione con votazione 105/110