



UNIVERSITÀ DEGLI STUDI DI PISA
Corso di Laurea Magistrale in Ingegneria Energetica

TESI DI LAUREA MAGISTRALE
**PROGETTAZIONE E SVILUPPO DI UNA LIBRERIA
ESEGUIBILE PER L'OTTIMIZZAZIONE DEL
RECUPERO TERMICO NEI TRENII DI SCAMBIO DI
UN IMPIANTO DI RAFFINAZIONE PETROLIFERA**

Relatori:

Prof. Ing. Claudio SCALI

Ing. Andrea ESPOSITO

Controrelatore:

Ing. Gabriele PANNOCCHIA

Candidato
Tommaso FERRARI

ANNO ACCADEMICO 2012-2013

Sommario

La tesi proposta riguarda lo studio e la risoluzione di alcuni aspetti inerenti al miglioramento di un sistema di controllo avanzato operante nella Raffineria di Livorno ENI. L'obiettivo di questo lavoro consiste nella realizzazione di una libreria (DLL) ATL COM utilizzabile da un server allo scopo di determinare in tempo reale (grazie all'acquisizione dei dati di campo) la distribuzione ottimale di varie portate di fluidi, frazionabili su diversi treni di scambio disposti in parallelo, al fine di ottenere il massimo recupero termico rispettando determinati vincoli imposti. I treni sono asserviti a un determinato processo di raffinazione, ovvero la distillazione di basi lubrificanti in una colonna tipo vacuum appartenente al ciclo di produzione del complesso industriale. Il sistema potrà essere applicato anche ad altri impianti della Raffineria dove siano presenti stazioni di scambio termico con caratteristiche affini. Il lavoro è stato svolto in collaborazione con il comparto di ricerca interno alla raffineria, nell'ambito di uno stage formativo. Nella realizzazione si è cercato di rispettare il più possibile tutte le esigenze sia teoriche che pratiche disposte dall'ente con cui vi è stata la collaborazione.

Indice

1	Introduzione	7
2	La Raffineria di Livorno: il processo di raffinazione del petrolio greggio	9
2.1	Descrizione generale	9
2.2	Schema generale del ciclo di produzione	13
3	Impianti di interesse per l'applicazione del sistema di ottimizzazione	16
3.1	Classe di problemi simili su cui applicare la libreria	16
3.2	La colonna di distillazione Vacuum e della stazione di scambio termico associata	17
3.2.1	Schema della colonna Vacuum	19
3.2.2	Caratteristiche della stazione di scambio termico	20
3.3	Gli altri impianti di interesse per l'applicazione della libreria OpTEsLib	24
4	Modello matematico e strategia di risoluzione	25
4.1	Formulazione matematica generale del problema	26
4.2	Caratteristiche della funzione obiettivo	27
4.2.1	Modello fisico matematico dell'unità scambiatore di calore	29
4.3	Funzione obiettivo ottenuta dalla linearizzazione	33
4.4	Caratteristiche dei vincoli	35
4.5	Strategia di risoluzione del problema vincolato	36
5	Analisi parametrica per la validazione del modello	38
5.1	Analisi delle conseguenze di eventuali errori sulle portate misurate	39
5.2	Considerazioni sugli altri parametri affetti da approssimazioni	43
5.3	Conclusioni dell'analisi parametrica	44

6	Architettura e implementazione del software	44
6.1	Struttura complessiva	45
6.1.1	Descrizione sintetica dei singoli file	46
6.1.2	Schema generale della classe ottimizzatrice	48
6.1.3	Esecuzione della libreria e differenti approcci in funzione del tipo di ottimizzazione	51
6.2	Modelli numerici e implementazione del codice	52
6.2.1	Implementazione delle funzioni associate all'unità scambiatore di calore	53
6.2.2	Algoritmo per la risoluzione di sistemi lineari e caratteristiche della fattorizzazione	56
6.2.3	Algoritmo e implementazione per la ricerca non vincolata del minimo globale	57
6.2.4	Algoritmo Nelder-Mead modificato per la ricerca vincolata del minimo	64
7	Schematizzazione del problema specifico e risultati numerici	65
7.1	Problematiche nell'utilizzo del software nell'applicazione sul problema di riferimento	69
7.2	Risultati dall'applicazione del software su dati storici registrati	71
7.2.1	Risultati relativi a un set di misure per la verifica del funzionamento generale	72
7.2.2	Verifica del recupero termico su più set di dati	77
8	Prospetto informativo per l'utilizzo della libreria OpTExLib	79
8.1	Funzioni per l'impostazione generale	82
8.1.1	Create_Class	82
8.1.2	Reset	83
8.1.3	Set_Parameters	83
8.1.4	Set_Obj	84
8.1.5	Set_Range	84
8.1.6	Initialize	85
8.2	funzioni di input	85

8.2.1	Set_Flows	85
8.2.2	Set_Fluid	86
8.2.3	Set_Connections	87
8.2.4	Set_Tin	87
8.2.5	Set_Tout	88
8.2.6	Set_Duty	88
8.3	Funzioni di output	89
8.3.1	Call_Flow	89
8.3.2	Call_Tout	89
8.3.3	(OPZIONALE) Call_Tout_initial	90
8.3.4	Call_Tin	90
8.3.5	Final_Enthalpy	90
8.3.6	Initial_Enthalpy	91
8.3.7	final_Tout	91
8.3.8	initial_Tout	91
8.3.9	Gain	92
8.3.10	Error_Duty	92
8.4	Ordine di utilizzo delle funzioni, esempio teorico	92
9	Conclusioni	94
A	Analisi di sensibilità parametrica	96
A.1	Analisi degli effetti dovuti alla variazione del parametro f conseguentemente al cambiamento delle condizioni operative	96
A.2	Analisi degli effetti dovuti alla variazione dei calori specifici in funzione del cambiamento delle condizioni operative . . .	98
A.3	Analisi delle conseguenze della variabilità del coefficiente globale di scambio termico in relazione alle diverse condi- zioni operative	99
A.4	Errori dovuti al calcolo di una temperatura incognita	100

1 Introduzione

Da come esposto negli ultimi rapporti annuali, Eni intende recuperare la redditività del settore Refining & Marketing nonostante la debolezza dello scenario internazionale e della situazione del mercato della raffinazione. Nella raffinazione, il graduale recupero verrà sostenuto dalla ottimizzazione delle attività industriali e di logistica mediante una maggiore flessibilità, integrazione ed efficienza dei processi. A tale scopo saranno attuati investimenti selettivi finalizzati all'incremento della capacità di conversione e dell'affidabilità degli impianti. Gli interventi in campo energetico e ambientale risultano ogni giorno sempre più dominanti tra le strategie di ricerca e sviluppo, oltre che essere imposti sempre più severamente dalle normative europee. Secondo questa linea nella Raffineria di Livorno sono in fase di studio e attuazione progetti per il miglioramento dei processi e della loro gestione, che possano garantire una sempre maggiore sostenibilità economica e ambientale. In particolare l'ottimizzazione e l'individuazione di condizioni ottimali nell'operatività dei processi gioca un ruolo importante. In questo quadro si inserisce il presente lavoro, che dovrebbe fornire risultati visibili nel risparmio di combustibili e conseguente maggiore economicità del processo.

Caratteristiche e obiettivi del progetto

Nello specifico questo lavoro di tesi riguarda la realizzazione di una libreria eseguibile dinamicamente denominata OpTE_xLib (Optimal Thermal Exchanges Library) sviluppata in C++ che provveda a svolgere determinate funzioni nell'ambito della ristrutturazione di un sistema di controllo avanzato DMCplus (Dynamic Matrix Control Corporation and Setpoint) operante nella gestione di vari processi della raffinazione. Il software consentirà di calcolare una nuova ripartizione delle portate di una stazione di scambio termico, asservita a determinati processi, la quale presenta un'organizzazione su varie linee in parallelo (treni di scambio), con l'obiettivo di ottenere la massimizzazione della temperatura di determinati flussi scelti - in aggiunta risulterà possibile impostare determinati flussi di calore

per alcune delle portate coinvolte. La massimizzazione dello scambio termico è stata operata nell'ordine di ottenere un risparmio di combustibile nel bilancio dei processi. Il software presenterà la possibilità di impostare determinati vincoli o parametri che dovranno essere rispettati dalla nuova condizione calcolata. Questa soluzione nasce da un problema determinato relativo a una struttura specifica. D'altra parte è stata posta grande attenzione per ottenere una generalità di applicazione. Infatti grazie all'inserimento della possibilità di modificare lo schema organizzativo dei componenti (scambiatori di calore - split, ovvero punti dove un determinato flusso viene frazionato in più parti) il software risulterà utilizzabile su una classe di problemi e strutture in cui si presentano le finalità suddette. La possibilità di eseguirlo come libreria permetterà di interfacciare gli input e gli output con altri *tool* (Excel, Visual Basic) in modo da avere una maggiore accessibilità e lettura dei dati e quindi attuare studi sistematici oltre che simulare e valutare altre modalità di organizzazione delle strutture. La strategia di utilizzazione prevede di eseguirlo ripetutamente nel corso del tempo, per sopperire a eventuali cambiamenti nelle condizioni operative (sporco degli scambiatori) e mantenere un assetto ottimizzato nel corso del tempo. Sarà possibile grazie a un'attenzione particolare della versatilità del *tool*, applicarlo in futuro su altre stazioni di scambio termico presenti nella Raffineria di Livorno. Si sottolinea che si è cercato di organizzare le funzioni di output e input in modo da rendere il più possibile intuitivi e semplici la configurazione e l'esercizio della libreria. È stata posta anche grande attenzione ai modelli utilizzati nel software che sono stati verificati mediante analisi parametriche e analisi dei risultati ottenuti dai test.

2 La Raffineria di Livorno: il processo di raffinazione del petrolio greggio

In questa sezione sarà esposto un'iniziale sintetica descrizione della Raffineria, del ciclo produttivo e dei processi coinvolti in essa, con particolare attenzione agli impianti nei quali potrà essere applicato il sistema di ottimizzazione.

2.1 Descrizione generale

La Raffineria di Livorno presenta una capacità di raffinazione primaria bilanciata di 84 mila barili/giorno e un indice di conversione dell'11% . Il grado o indice di conversione rappresenta la capacità di minimizzare la produzione di prodotti pesanti come l'olio combustibile, a vantaggio di prodotti leggeri come benzina e gasolio. Più è alto questo parametro più la raffineria presenta flessibilità e offre maggiori prospettive di redditività, d'altra parte un grado elevato prevede una maggiore complessità e consistenti investimenti per le strutture. La Raffineria di Livorno nasce in collegamento con il distretto portuale, sia per la possibilità di rifornire direttamente il porto con i combustibili ottenuti che per lo sfruttamento di esso come vettore adibito al trasporto dei prodotti. Negli anni si sono susseguiti vari interventi, fra i principali realizzati vi è stata il miglioramento della produzione di basi lubrificanti e paraffine (AgipPlas AGIP - Produzione Lubrificanti Additivi e Solventi), e l'ampliamento e ammodernamento della centrale di produzione di energia elettrica (199 MW) e termica (695 MW) Eni Power (fornisce energia anche al di fuori della raffineria) che utilizza olio combustibile, gas di raffineria e gas naturale. Nella raffineria di Livorno attualmente sono prevalentemente prodotti gpl, benzine, gasoli, olio combustibile per bunkeraggi (sempre più in minori quantità) e basi lubrificanti. La produzione di lubrificanti è attuata in due linee dedicate di cui solo una attualmente in servizio. Il greggio prevalentemente utilizzato in precedenza era di tipo pesante, negli ultimi anni invece si è spostato l'utilizzo su greggi di qualità pregiata atti ad ottenere prodotti di qualità e redditività migliori. Con ciò è aumentato il grado di conversione, nonostante si attest

ancora su valori bassi a causa della mancanza di strutture adeguate per una raffinazione più fine. In generale gli investimenti attuati si inseriscono in un processo continuo di adeguamento tecnologico, apportando miglioramenti agli impianti di produzione anche e soprattutto dal punto di vista della garanzia delle condizioni di sicurezza e di rispetto dell'ambiente. L'attuale organizzazione della raffineria è visibile in figura 2.2.

Il collegamento con le strutture portuali di Livorno è garantito dalla presenza di autostrade, ferrovie e un oleodotto, mentre i depositi di Firenze sono connessi attraverso due oleodotti consentendo così di ottimizzare le attività di ricezione, movimentazione e distribuzione dei prodotti. Oltre a queste infrastrutture sono presenti due darsene interne grazie alle quali avviene l'approvvigionamento del greggio.

Nel corso degli anni è costantemente cresciuto l'utilizzo di gas naturale in alternativa all'olio combustibile in ordine al funzionamento della raffineria. Questa scelta è chiaramente dettata dal minor impatto ambientale della combustione. Va sottolineato che il gas naturale è un combustibile pregiato con ciò può rivelarsi importante l'attuazione di strategie volte al risparmio o all'uso più razionale possibile di questa risorsa (questo si collega anche con l'economia dell'impianto). Attualmente sono in corso studi e applicazioni per migliorare la qualità e il risparmio energetico della raffineria. Il basso indice di conversione può suggerire l'opportunità di attuare interventi per il miglioramento dei processi e degli impianti con margini consistenti. Gli interventi possono naturalmente riguardare la struttura, le componenti degli impianti oltre che il controllo e la gestione degli stessi. In questi ultimi tipi di interventi si concretizza il sistema di controllo avanzato di cui fa parte questo progetto; nello specifico e in prima istanza esso riguarda la colonna di distillazione *vacuum*, attualmente a regime, per la produzione di basi lubrificanti. Come già esposto nell'introduzione il controllo dovrebbe consentire di ottenere un maggiore risparmio energetico conservando la qualità dei prodotti. La possibilità di ottenere questo obiettivo è collegata all'utilizzo dei risultati provenienti da un altro progetto di tesi, come sarà maggiormente chiarito in seguito.



Figura 2.1: vista della Raffineria di Livorno.

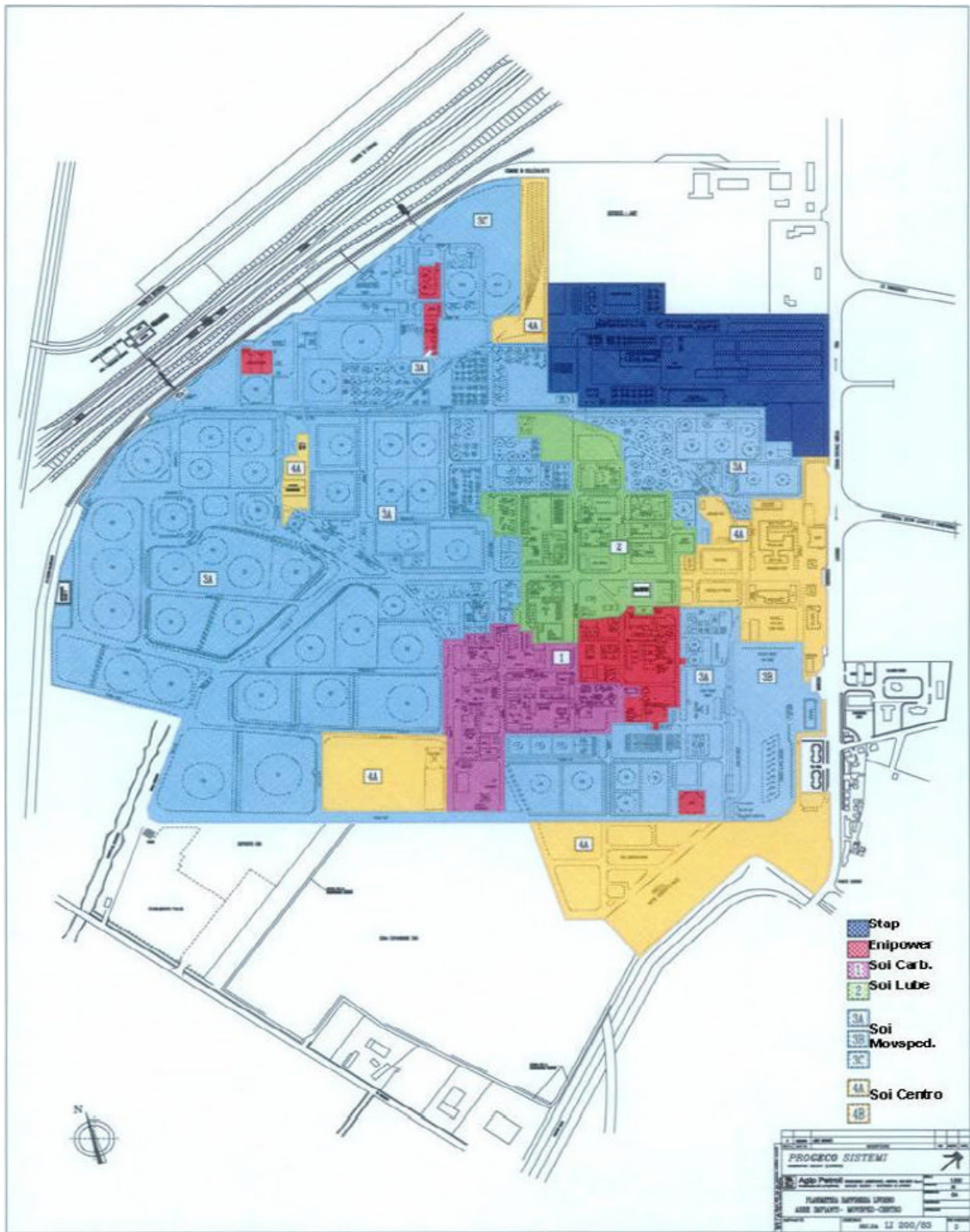


Figura 2.2: planimetria del sito e della Raffineria con evidenziati i vari settori.

2.2 Schema generale del ciclo di produzione

Le raffinerie presentano strutture in cascata, i residui provenienti dalle prime distillazioni vengono utilizzati come alimentazione degli impianti successivi, fino ad ottenere determinati prodotti o frazioni con le caratteristiche volute. Nel caso di Livorno lo schema generale è riassunto nella figura 2.3. Di particolare rilevanza risultano gli impianti evidenziati dalle cornici rosse, su cui sarà approfondito in seguito. Il ciclo produttivo inizia con la colonna di distillazione Topping in cui convoglia il greggio e da cui si ottengono in ordine di altezza di estrazione decrescente:

- gas di raffineria ($C_1 - C_2, H_2 \dots$);
- GPL ($C_3 - C_4$);
- benzine (C_5 , idrocarburi basso-bollenti ($<150^\circ$));
- cherosene ($C_{10} - C_{14}$);
- gasolio ($C_{15} - C_{30}$);

Prima di poter ottenere questi prodotti in realtà sono presenti trattamenti secondari delle frazioni, caratterizzati da vari processi (desolforazione, *splitting*, *platforming*), per ottenere i prodotti, le purezze e le caratteristiche cercate. Il residuo in uscita da questa prima colonna (RA, residuo atmosferico o residuo *topping*) viene in parte inviato direttamente, in parte stoccato, alla colonna Vacuum adibita alla produzione specifica di basi lubrificanti e paraffine. Quest'ultima, poiché oggetto specifico del progetto trattato, verrà descritta nel dettaglio nella sezione successiva. In connessione con quanto sarà esposto nel paragrafo successivo si può dire che il residuo di quest'ultima colonna RVC viene inviato in un impianto PDA (Propane DeAsphalting) da dove si ottiene una parte ancora utilizzabile per la produzione di basi lubrificanti più pesanti e un residuo pesante composto da asfalti (bitume). In quest'ultimo processo, come nei due precedenti vi è la necessità di consistenti scambi di calore e recupero energetico.

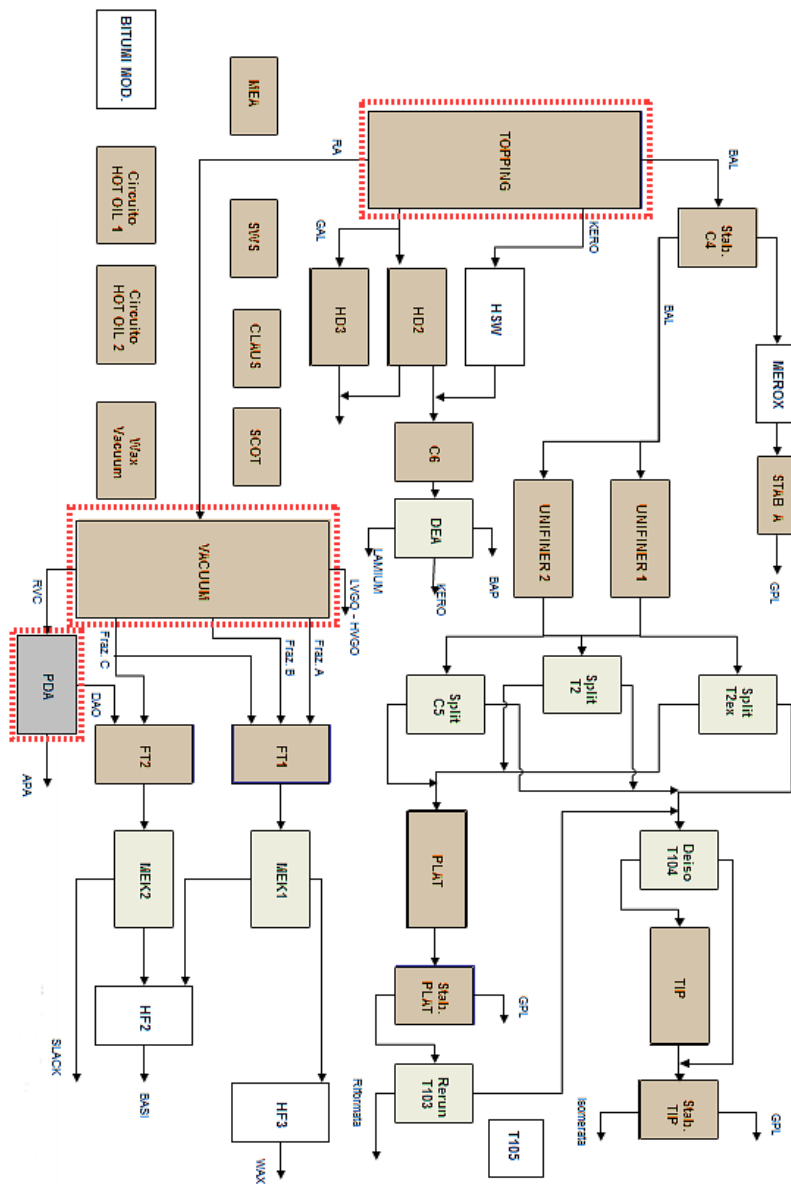


Figura 2.3: schema del ciclo di produzione della Raffineria di Livorno.

Si riportano sinteticamente gli impianti principali che costituiscono il ciclo di produzione in relazione allo schema precedente:

- colonna Topping (Distillazione Atmosferica);

- produzione GPL: impianti stabilizzazione benzine (stab.benz C4), Merox GPL e stabilizzazione GPL (stab. GPL);
- impianto di desolforazione catalitica Hydrosweeting (HSW);
- produzione benzine: impianti Unifiner 1 e 2 (UNI 1 / 2), colonne distillazione Splitter T2, T2 ex, C5, isomerizzazione (ISO) e Platforming (PLAT);
- produzione solventi dearomatizzati (impianto DEA - dearomatizzazione), produzione di un taglio pesante di benzina detta BAP, Kerosene, Lamium (particolare prodotto di raffineria);
- produzione di gasoli (impianti di desolforazione catalitica Hydrofiner (HD 2 / 3));
- colonna Vacuum (distillazione sotto vuoto VPS);
- 2 strutture di assorbimento con MEA (miscele ammine terziarie), per il lavaggio e la purificazione dall'H₂S del gas di risulta;
- 2 impianti di recupero dello zolfo (Claus), che convertono l'H₂S;
- 1 impianto SCOT (Shell Claus Off gas Treatment) ;
- impianto di deasfaltazione al propano PDA (Propane DeAsphalting);
- impianti di raffinazione al furfurolo (FT1 - FT2);
- impianti di Deparaffinazione con Solvente (MEK1 - MEK2);
- impianto Hydrofinishing 2 (HF2);
- Impianto Hydrofinishing 3 (HF3).

3 Impianti di interesse per l'applicazione del sistema di ottimizzazione

In questa sezione si descriverà inizialmente la classe di problemi trattata e le modalità di applicazione del software. In seguito verranno descritti gli impianti che presentano i requisiti esposti e in particolare l'impianto di riferimento per le simulazioni e la prima applicazione effettiva.

3.1 Classe di problemi simili su cui applicare la libreria

La libreria costituita si riferisce a una determinata classe di problemi. Il software può essere applicato in situazioni in cui siano presenti necessariamente queste condizioni e obiettivi richiesti:

- presenza di stazioni di scambio termico che prevedano la misura e il controllo delle portate volumetriche o massiche (presenza di un controllo avanzato, buona precisione delle misure);
- presenza di reti di scambiatori (a superficie, tipo *shell and tube*) con più treni di scambio paralleli - attualmente la libreria può operare su sistemi con due treni di scambio ma vi è la possibilità con contenute modifiche di applicarla anche a sistemi con un numero indefinito di treni (nella Raffineria di Livorno si presentano solo casi con due treni);
- numero contenuto di scambiatori presenti (<50) ma sufficienti da giustificare l'approccio numerico (>3) ;
- funzionamento in regime stazionario del sistema;
- presenza di misuratori di temperatura all'ingresso e all'uscita degli scambiatori (almeno in larga parte) con una sufficiente precisione;
- obiettivo consistente nel voler massimizzare il recupero termico mediante la massimizzazione dell'entalpia in uscita da determinati flussi scelti,

operando sul frazionamento nei diversi treni di scambio delle portate totali coinvolte;

- necessità di operare sul sistema in evoluzione (eseguire periodicamente il software), in collegamento con il cambiamento della qualità dei flussi o dello sporcamento degli scambiatori, in modo da correggere nel tempo le variabili;
- sistemi di cui non si conoscano con precisione le proprietà fisiche dei fluidi;
- necessità di ottenere determinati flussi di calore (da qui in poi saranno denominati impropriamente ma comodamente *duties*) per alcune portate in collegamento all'eventuale processo a cui è asservita la stazione di scambio termico (opzionale);
- necessità di monitorare l'evoluzione degli scambiatori nel tempo (opzionale).

Queste caratteristiche verranno maggiormente approfondite in seguito, e in connessione ad esse saranno esposte le scelte operate nella realizzazione della libreria.

Attualmente nella raffineria di Livorno vi sono tre stazioni di scambio termico che presentano queste caratteristiche. Nel prossimo paragrafo sarà descritto il caso di riferimento trattato nello specifico, in seguito saranno brevemente descritti gli altri casi di interesse soffermandosi solo sulle caratteristiche degli impianti in maniera generale. Sul caso di riferimento sono sette effettivamente attuate le simulazioni con dati storici ed è stato testato sul server il software con le misure di campo istantanee.

3.2 La colonna di distillazione Vacuum e della stazione di scambio termico associata

Il caso di riferimento per l'applicazione iniziale del software è la colonna di distillazione Vacuum. La scelta di questo impianto è connessa alla necessità specifica di

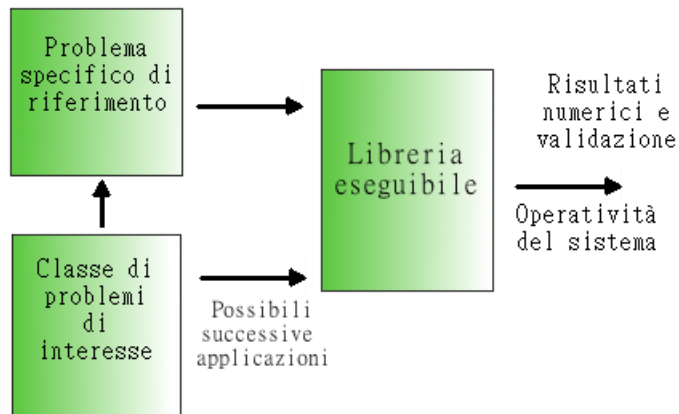


Figura 3.1: schema di applicazione del software

dover garantire determinati flussi di calore (conosciuti) su alcune frazioni estratte dalla colonna, e quindi utilizzare a pieno le potenzialità dello strumento. I flussi di calore imposti saranno da qui in poi denominati

La colonna di distillazione per la produzione di basi lubrificanti opera sottovuoto (tipo *vacuum*). Questa modalità di raffinazione presenta notevoli vantaggi, infatti consente di recuperare dal residuo *topping* (o atmosferico), proveniente dalla colonna di distillazione omonima, ulteriori prodotti che presentano temperature di vaporizzazione così elevate da poter causare scissioni dei composti nel caso si operasse in condizioni atmosferiche. Infatti operando sottovuoto (pressione a 0.05 bar) e tramite l'aggiunta di vapore nelle serpentine dei forni, dove avviene la fase di riscaldamento, si abbassa la pressione parziale degli idrocarburi da vaporizzare e di conseguenza la temperatura di vaporizzazione. I distillatori di tipo *vacuum* richiedono solitamente temperature più elevate all'uscita dei forni rispetto a quelli atmosferici, ciò può comportare maggiori perdite di calore verso l'esterno durante il processo. Uno dei fattori importanti nella distillazione di basi lubrificanti è certamente la qualità dei prodotti. Essi infatti devono presentare delle caratteristiche chimico-fisiche anche stringenti e connesse con il loro impiego. Ad essa è asservita una stazione di recupero termico che appunto consente l'applicazione

del software.

3.2.1 Schema della colonna Vacuum

L'impianto di distillazione sotto vuoto (esclusa la stazione di scambio termico) è rappresentato in figura 3.2. Il residuo della distillazione atmosferica è inviato al polmone V203 dalle pompe del fondo colonna dell'impianto Topping. Mediante un sistema di controllo e l'attuazione su delle pompe boosters che aspirano dai serbatoi di RA di integrazione, si garantisce il livello del battente nel polmone di carica V203. A monte del V203 la carica si preriscalda in E255 scambiando con vapore alla pressione di rete $2,5 \text{ kg/cm}^2$. Aspirata dalle pompe P201 A e C, la carica è inviata a preriscaldarsi nei due treni di scambio, sotto il controllo degli strumenti di portata. Una volta preriscaldata, la carica entra negli otto rami dei forni F201 ed F202, ove raggiunge la temperatura necessaria per la distillazione. Negli ultimi serpentini della zona radiante dei forni è prevista una iniezione di vapore ad alta-media pressione per diminuire il tempo di contatto e ridurre la formazione di coke. La carica, lasciati i forni, entra nella zona flash della colonna C201. I vapori leggeri salgono verso la testa mentre il residuo liquido scende verso il fondo colonna. Quasi la totalità dei vapori che salgono, tornano a condensarsi sui piatti, in funzione della temperatura di ebollizione degli idrocarburi che li costituiscono. Solo una piccola parte di idrocarburi più leggeri ed incondensabili sono aspirati dal sistema di eiettori a vapore, i quali mantengono il vuoto necessario all'interno della colonna. Il condensato della mandata degli eiettori si raccoglie nel separatore V202 dove l'acqua si separa e, per mezzo di un controllore di livello differenziale, viene inviata all'impianto di trattamento acque reflue, mentre gli idrocarburi che superano il setto interno ricadono nell'altro settore del polmone e sono inviati allo slop. Gli idrocarburi incondensabili che escono dal 3° stadio eiettori, sono inviati alla combustione nei forni F201 ed F202. Il bilancio termico della C201 è assicurato dai Pumparounds (P.A.). Dalla parte superiore della colonna vengono prelevati gasoli che verranno inviati allo stoccaggio (VGO). Ad altezza inferiore (25° piatto) viene prelevato il taglio "A". Una parte passa allo

stripper C202/A dove viene strappato con vapore surriscaldato. L' altra parte del taglio "A" è prelevata a monte dallo stripper, inviata al treno di scambio e successivamente reimpressa in colonna ad altezza superiore a quella di estrazione. Ancora a minore altezza (19° piatto) viene prelevato il taglio "B". Una parte passa allo stripper C202/B dove subisce lo stesso strippaggio del taglio "A". Il fondo stripper viene inviato ai generatori di vapore E215 ed E230, ai refrigeranti E212 ed E227, ed infine a stoccaggio. Lo strip-out di testa rientra in colonna sempre sopra al livello di estrazione. L' altra parte del taglio "B", aspirato a monte dello stripper, è inviato ai treni di scambio e quindi riciclato in colonna sopra il piatto di estrazione. Dall'8° piatto viene prelevato il taglio "C". Una parte passa allo stripper aspirato dalla pompa P-208 A/B ed è inviata agli scambiatori successivamente raffreddata in E209/E224 e inviata a stock. Lo strip-out rientra in colonna sopra l'undicesimo piatto. L' altra parte del taglio "C" viene reintrodotta in colonna sopra il piatto 8° come bottom P.A., dopo aver scambiato calore nei treni. Nella parte inferiore della colonna vengono estratte direttamente paraffine (WAX) che vengono inviate allo stoccaggio. Il residuo liquido della distillazione separatosi dai vapori nella zona flash, si raccoglie sul fondo colonna, dopo aver subito uno strippaggio con vapore surriscaldato viene inviato anch'esso all stazione di scambio. Dopo i generatori di vapore E213 ed E228, il residuo va a stoccaggio, raffreddato da refrigeranti ad acqua. A monte dell'ultimo refrigerante è prevista una diluizione con petrolio o gasolio per abbassare la viscosità e la temperatura di collaggio. Una parte di residuo in uscita viene riciclata al fondo colonna con lo scopo di raffreddare la massa liquida e ridurre la formazione del carbone.

3.2.2 Caratteristiche della stazione di scambio termico

La stazione di scambio termico associata alla colonna ha visto vari rivoluzionamenti nel corso del tempo. Inizialmente infatti era presente solo un treno di scambio, successivamente l'esigenza di avere una maggiore capacità produttiva ha portato all'introduzione di un altro treno parallelo. Anche i collegamenti tra i vari scambiatori dei diversi treni sono stati modificati secondo esigenze di ottimizza-

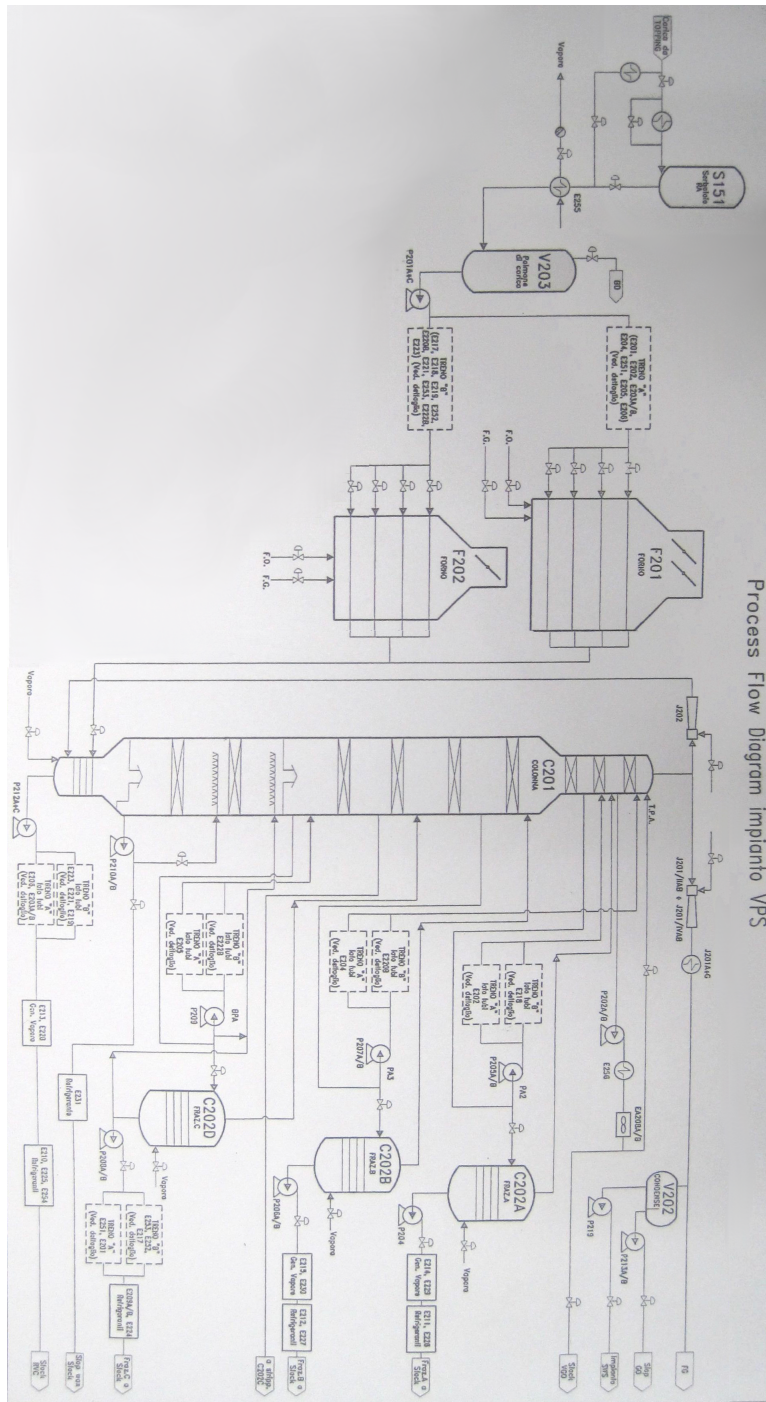


Figura 3.2: schema della colonna di distillazione Vacuum senza la stazione di scambio termico.

zione e maggiore produttività. La libreria realizzata è stata quindi concepita nella modalità in cui possa fornire **la possibilità di modificare lo schema analizzato**, evitando di dover ricompilare il codice (con le difficoltà connesse alla comprensione dello stesso da parte di utenti che hanno accesso solamente alle informazioni di base - questa tesi fornisce anche il prospetto di utilizzo della libreria con le funzioni annesse nella sezione 8). Lo schema attualmente presente è riportato in figura 3.3. La stazione di scambio termico è composta da 17 scambiatori, la carica entra da sinistra e si divide nei due treni, andando verso destra incontra i flussi delle frazioni provenienti dalla colonna C201. Come si può osservare le tre frazioni A, B, C (la parte non passante per lo stripper), provenienti dai Pumparounds (PA2, PA3, BPA) scambiano solamente con uno scambiatore per ogni treno (nel treno inferiore con 1, 3, 5 - nel treno superiore con 8, 11, 14), al contrario le restanti frazioni intersecano più volte le linee della carica. Da notare come la parte Bottom C intersechi più volte entrambi i treni e venga splittata prima di raggiungere i primi scambiatori. Lo schema verrà utilizzato in ambito del modello matematico e dell'architettura del programma.

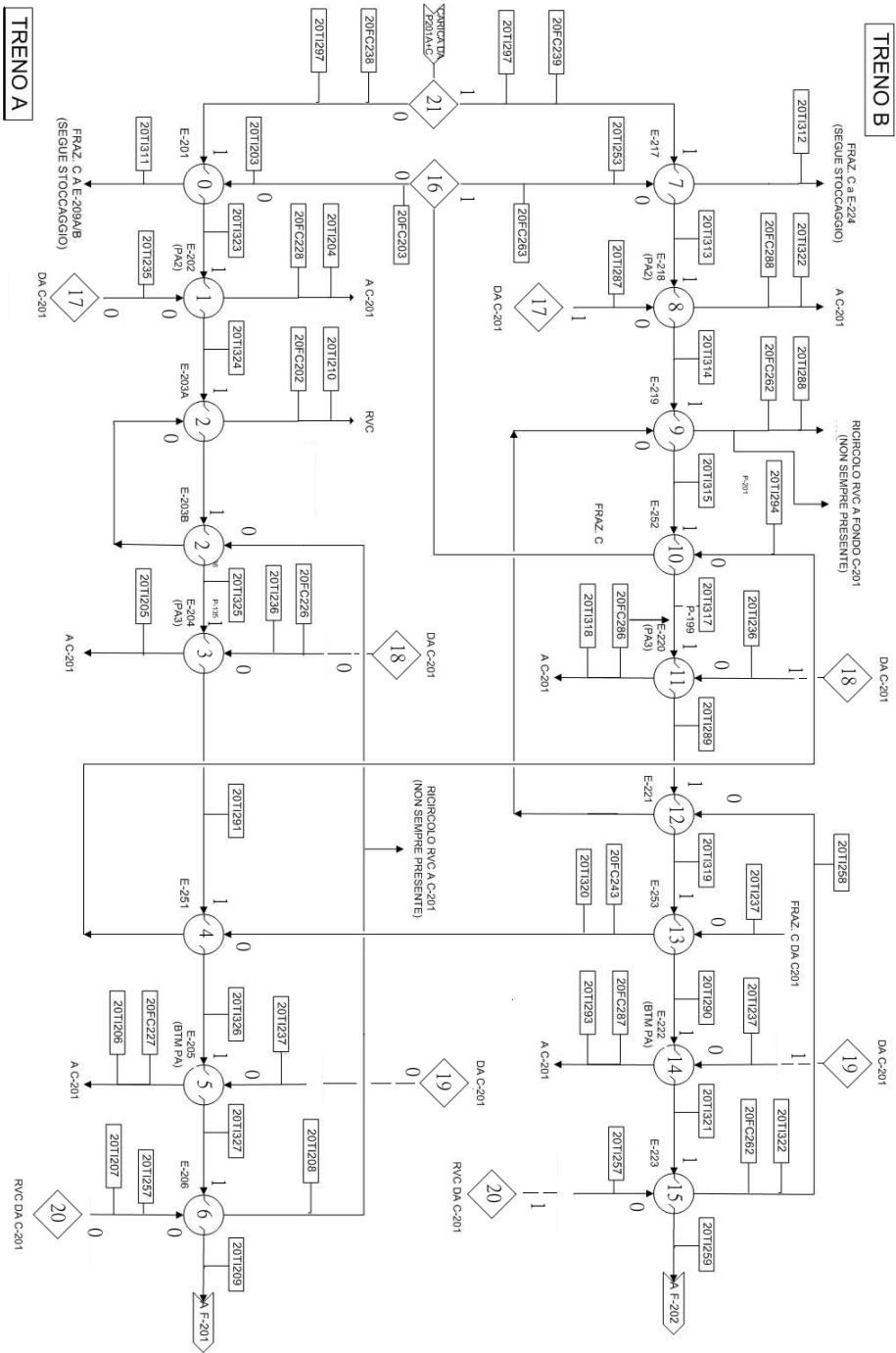


Figura 3.3: schema della stazione di scambio termico

3.3 Gli altri impianti di interesse per l'applicazione della libreria OpTeXLib

Oltre alla colonna Vacuum sono presenti altri due impianti a cui è possibile applicare il software. Essi sono la colonna Topping e la colonna PDA (Propane DeAsphalting). Infatti a questi sono asservite delle stazioni di scambio termico costituite da diversi scambiatori di calore connessi fra loro. Per entrambe le colonne i treni contribuiscono a riscaldare la carica e a raffreddare le frazioni estratte (che in caso sono riciclate), quindi garantiscono il recupero termico delle numerose evaporazioni e condensazioni subite dalle sostanze coinvolte. Poiché lo scambio di calore avviene su treni in parallelo risulta possibile operare sui gradi di libertà presenti (frazione di split del flusso totale, ad esempio quanto inviare ad un treno o all'altro o a un terzo) per ottenere il risultato di recupero termico massimizzato nella carica. Il processo di de-asfaltazione prevede che la carica risalendo la colonna venga in contatto in controcorrente con un solvente (nel caso specifico propano) che provvede a disciogliere i composti alifatici separandoli dagli asfalteni. Questi due impianti quindi saranno presi in considerazione nell'eventuale applicazione di questa libreria con eventuali, ma contenute modifiche del codice attuale (progetto successivo).

4 Modello matematico e strategia di risoluzione

In questa sezione inizialmente si esporrà la trattazione matematica del problema, le caratteristiche della funzione obiettivo il modello utilizzato per gli scambiatori, quindi si esporrà la strategia risolutiva poi tradotta in ambito numerico. La modellizzazione sarà compresa più a fondo nella sezione successiva, in cui sarà applicata al caso di riferimento.

I sistemi trattati dal software sono reti di scambiatori e split presumibilmente disposti su più treni (come esempio si faccia riferimento allo schema 3.3 a pagina 23). Le variabili su cui si vuole agire per ottenere la massimizzazione della funzione obiettivo consistono nei gradi di frazionamento applicati agli split. In essi valgono le seguenti assunzioni:

$$m_{t,i} = m_{0,i} + m_{1,i} \text{ con } i = 0, 1 \dots n_{split} \quad (4.1)$$

dovuta alla conservazione della massa in regime stazionario ($m_{t,i}$ è la massa totale) e

$$x_i = m_{0,i}/m_{t,i} \quad (4.2)$$

ovvero la definizione di grado di frazionamento. Il grado di frazionamento è chiaramente compreso tra 0 e 1. Si sono indicate con m le portate massiche coinvolte nello split, come si può osservare nella figura seguente.

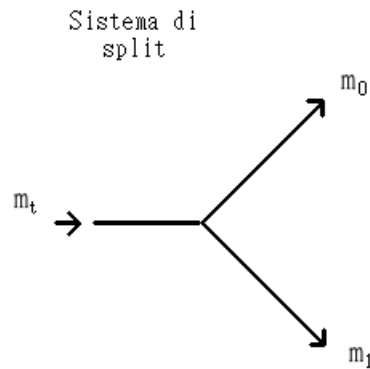


Figura 4.1: schema di uno split

4.1 Formulazione matematica generale del problema

Il problema consiste in una ottimizzazione vincolata, ovvero data una funzione obiettivo e determinati vincoli ci si propone di trovare un vettore composto dagli x_i per cui risulti:

$$x_s : \begin{cases} E(x_s) = \min[E(x)] & \text{con } x_s \in \Omega \subset \mathbb{R}^n \\ g_j(x_s) \leq 0 & j = 1, \dots, m_1 \\ y_k(x_s) = 0 & k = 1, \dots, m_2 \end{cases} \quad (4.3)$$

Per la precisione questo problema è risolvibile mediante un metodo che è la generalizzazione del metodo dei moltiplicatori di Lagrange, ovvero le condizioni di Karush–Kuhn–Tucker, verificate se presenti determinate condizioni di regolarità dei vincoli, alcune più stringenti altre più deboli (si rimanda a testi specialistici). Questo problema può risultare non banale soprattutto se non si ha una conoscenza analitica della funzione E . La verifica delle condizioni di regolarità in essa può presentare grosse difficoltà, a questo si sommano problematiche nella risoluzione numerica. Come si vedrà nella schematizzazione del sistema di riferimento vi sarà

la necessità di imporre solo condizioni vincolanti di disuguaglianza (funzioni g_j), questo semplifica in parte il problema:

$$x_s : \begin{cases} E(x_s) = \min[E(x)] \text{ con } x_s \in \Omega \subset \mathbb{R}^n \\ g_j(x_s) \leq 0 & j = 1, \dots, m \end{cases} \quad (4.4)$$

4.2 Caratteristiche della funzione obiettivo

Dati una serie di valori per le componenti di \bar{x} ad esso corrisponderà un vettore di portate massiche m involventi nel sistema. Le proprietà fisiche dei flussi di materia in ingresso si presentano determinate. Quindi **la distribuzione delle portate totali determina le proprietà fisiche in uscita dal sistema** (non vi sono altri cambiamenti se non le portate). Come si vedrà le proprietà fisiche di interesse per il calcolo della funzione obiettivo sono la temperatura, i calori specifici a pressione costante e la densità se si tratteranno le portate volumetriche (ragionevolmente il ruolo della pressione viene trascurato trattandosi di liquidi). Le due ultime proprietà sono comunque funzioni della temperatura, quindi le uniche variabili **incognite significative risultano appunto alcune delle temperature in uscita** che hanno semplicemente il ruolo di funzioni intermedie.

La risoluzione delle nuove condizioni di uscita è data dalla risoluzione di un sistema (presumibilmente non lineare) le cui equazioni sono fornite dal modello utilizzato per gli scambiatori. Ogni scambiatore può essere visto come un sistema a se stante (equazioni caratteristiche indipendenti dagli altri scambiatori). Queste equazioni sono determinate dai parametri in ingresso allo scambiatore e determinano le temperature in uscita. Quindi conosciute le variabili in ingresso ogni scambiatore presenta un sistema non lineare con due equazioni (deve essere univocamente determinato) e due incognite, ovvero le due temperature di uscita riferite ai due flussi circolanti. I parametri in ingresso sono 4 (2 portate e 2 temperature - gli effetti della pressione sono trascurati). Il sistema può essere quindi così espresso:

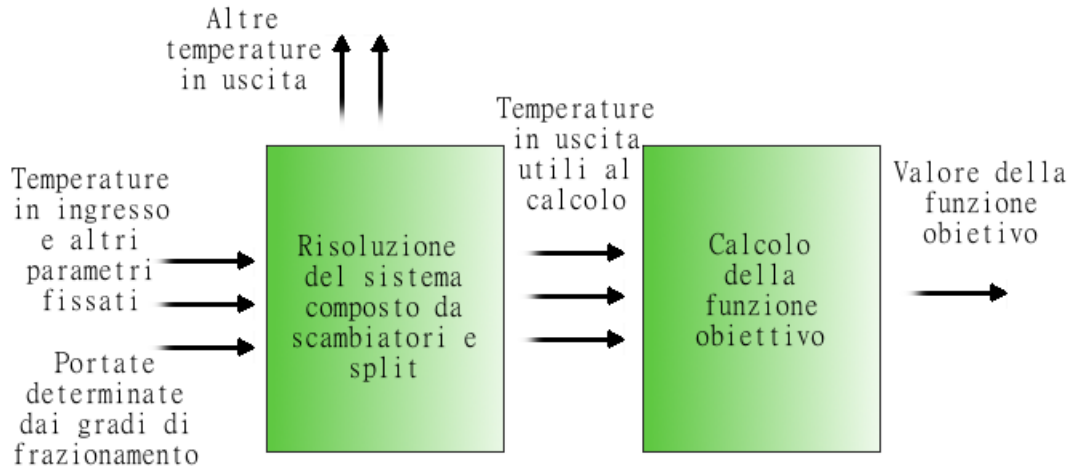


Figura 4.2: schema del calcolo della funzione obiettivo

$$\begin{cases} f_{0,i}(T_{0,i}^{out}, T_{1,i}^{out}, m_{0,i}, m_{1,i}, T_{0,i}^{in}, T_{1,i}^{in}) = 0 \\ f_{1,i}(T_{0,i}^{out}, T_{1,i}^{out}, m_{0,i}, m_{1,i}, T_{0,i}^{in}, T_{1,i}^{in}) = 0 \end{cases} \quad (4.5)$$

dove i identifica il determinato scambiatore, le incognite come si è detto sono $T_{0,i}^{out}, T_{1,i}^{out}$. Gli indici 0 e 1 rappresentano le linee dello scambiatore. Conoscendo le connessioni, le portate si possono sostituire nei sistemi di ogni scambiatore. Le temperature in ingresso invece dipendono dalla risoluzione degli scambiatori connessi o se non vi è connessione dai valori in ingresso al sistema complessivo. Quindi dato un certo numero di scambiatori le variabili del sistema complessivo atto a risolvere tutti gli scambiatori risultano due volte questo valore. Dopo l'esposizione del modello adottato per la singola unità scambiatore si vedrà come il sistema di equazioni non lineare viene approssimato a un sistema lineare.

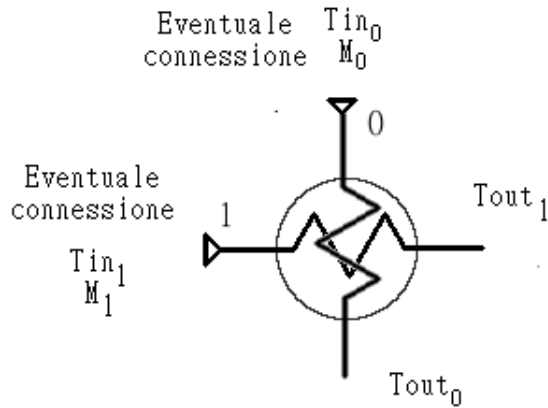


Figura 4.3: schema unità scambiatore

4.2.1 Modello fisico matematico dell'unità scambiatore di calore

Nella sostanza ogni singolo scambiatore è un'entità a se tale per cui date le temperature e le portate in ingresso in situazione di regime stazionario darà luogo univocamente a determinate temperature di uscita. In questa sottosezione si individuerà il modello opportuno per la modellizzazione della singola unità. Esistono diversi approcci nella modellizzazione degli scambiatori a superficie. Essi dipendono chiaramente dalla struttura dello scambiatore e dalla modalità dei flussi. Rigorosamente sarebbe necessario per determinati flussi di materia e quindi velocità, temperature, pressioni locali e le correlate proprietà termodinamiche nel determinato stato, determinare il flusso di calore in modo infinitesimale e quindi integrare su tutto lo scambiatore. Spesso è possibile in ambito ingegneristico l'utilizzo di modelli semplificati che si servono di parametri macroscopici. Essi possono provvedere al dimensionamento dello scambiatore o alla predizione di determinate variabili come le portate e le temperature. Nel caso in questione è necessario ottenere le temperature in uscita date le condizioni di ingresso nelle due linee (mantello e fascio tubiero) fissate le condizioni di temperatura in ingresso e le portate, variare rispetto alla condizione iniziale di funzionamento. Si è scelto di

utilizzare il modello LMTD (*Log Mean Temperature Difference*) che fa riferimento alla differenza di temperatura media logaritmica poiché come si vedrà si presta in maniera ottimale alla risoluzione del problema. Il modello non tiene conto delle perdite termiche verso l'esterno (adiabaticità), ma è una condizione ragionevole poiché gli scambiatori sono isolati all'esterno e non presentano grandi superfici esterne paragonate alle superfici di scambio. Per comodità ci si riferisce alle linee identificandole con il fluido cedente calore H e il fluido assorbente C percorrenti le stesse. Il modello si appoggia sull'approssimazione dello scambiatore a uno scambiatore teorico con flussi monodimensionali perfettamente in controcorrente, considerando le temperature di Bulk. Si può quindi descrivere mediante le equazioni:

$$dQ = U (T_H - T_C) dS \quad (4.6)$$

$$dQ = c_{p,C} m_C dT_C \quad (4.7)$$

$$dQ = -c_{p,H} m_H dT_H \quad (4.8)$$

In cui:

- dQ è il flusso di calore infinitesimo scambiato;
- U è il coefficiente di scambio termico riferito alla parte di area di scambio;
- dS è l'area infinitesima di scambio;
- T_c, T_h sono le temperature rispettivamente del del fluido freddo e del fluido caldo;
- m_c, m_h le portate rispettivamente del fluido freddo e del fluido caldo;
- $c_{p,c}, c_{p,h}$ i calori specifici medi a pressione costante rispettivamente del fluido freddo e del fluido caldo (si accetta l'errore introdotto dall'uso di quan-

tità medie costanti, ciò risulta comunque valido per salti di temperatura non eccessivi);

Considerando il coefficiente di scambio e i calori specifici costanti si possono combinare le equazioni ed integrare ottenendo in termini globali:

$$\begin{cases} c_{p,c}m_c(T_{c,out} - T_{c,in}) = c_{p,h}m_h(T_{h,in} - T_{h,out}) = Q & \text{conservazione dell'energia (approssimata)} \\ Q = SU LMTD = SU \frac{(T_{H,in} - T_{C,out}) - (T_{H,out} - T_{C,in})}{\ln \frac{T_{H,in} - T_{C,out}}{T_{H,out} - T_{C,in}}} \end{cases} \quad (4.9)$$

Risulta chiaro che ciò che è stato esposto è un modello approssimato, oltretutto poiché gli scambiatori spesso non presentano un flusso effettivamente in controcorrente si introduce un fattore correttivo sulla *LMTD* sostituendo questa con un altro parametro *MTD* (differenza di temperatura media):

$$MTD = LMTD * f \quad (4.10)$$

Il parametro *f*, che assume valori minori di 1 (si aggira attorno a 0.8-0.9 per gli scambiatori shell and tube) introduce un effetto riduttivo sulla differenza di temperatura media logaritmica, ragionevolmente collegato al regime non perfettamente in controcorrente.

Il suddetto modello verrà comunque utilizzato in una determinata modalità ora esposta. Come si può vedere conoscendo il flusso di calore si può calcolare *U* che sintetizza l'informazione riguardante la capacità del sistema di scambiare calore in un singolo parametro. Esso è calcolato mediante la differenza di temperatura media logaritmica, quindi è un parametro medio dipendente dalle temperature di ingresso e di uscita. Utilizzando le equazioni 4.9 si può quindi riuscire a calcolare le temperature di uscita:

$$A = \frac{SU}{c_{p,C}m_C} \quad B = \frac{SU}{c_{p,H}m_H} \quad (4.11)$$

$$\left\{ \begin{array}{ll} a_1 = \frac{B-A}{Be^{B-A}-A} & \text{con } A \neq B \\ a_1 = \frac{1}{B+1} & \text{con } A = B \\ a_2 = \frac{Ae^{A-B}-A}{Ae^{A-B}-B} & \text{con } A \neq B \\ a_2 = \frac{B}{B+1} & \text{con } A = B \\ b_1 = \frac{Be^{B-A}-B}{Be^{B-A}-A} & \text{con } A \neq B \\ b_1 = \frac{B}{B+1} & \text{con } A = B \\ b_2 = \frac{A-B}{Ae^{A-B}-B} & \text{con } A \neq B \\ b_2 = \frac{1}{B+1} & \text{con } A = B \end{array} \right. \quad (4.12)$$

$$\left\{ \begin{array}{l} T_{H,out} = a_1 T_{H,in} + b_1 T_{C,in} \\ T_{C,out} = a_2 T_{H,in} + b_2 T_{C,in} \end{array} \right. \quad (4.13)$$

conoscendo i parametri U , c_p , m . Il sistema è lineare ed è determinato da solo due parametri A e B per evidenti condizioni di simmetria (lo scambiatore presumibilmente funziona in modo identico se si dovessero invertire i due fluidi circolanti, ovvero far passare il fluido caldo nel mantello invece che nei tubi e viceversa). Ora supponendo di voler utilizzare invece una situazione di funzionamento di uno scambiatore, in cui si conoscono portate e temperature sia di uscita che di ingresso e di determinare il sistema. Quindi si procede a calcolare A e B e risulta che essi sono dati da:

$$A = \frac{T_{C,out} - T_{C,in}}{LMTD} \quad B = \frac{T_{H,in} - T_{H,out}}{LMTD} \quad (4.14)$$

che sostituite nel sistema danno chiaramente luogo a un'identità. Ora supponiamo di attuare una perturbazione contenuta delle portate (mantenendo costanti le temperature in ingresso) e di poter considerare valide ipotesi:

1. variazione contenuta di U : effettivamente esso varia con le velocità (e quindi le portate) ma si si aspetta ed è stato verificato numericamente che non

si discosti in modo eccessivo (oltretutto una variazione contenuta non ha effetti marcati sulle temperature in uscita);

2. anche se le temperature in uscita variassero i calori specifici calcolati mediamente risulterebbero circa costanti (dipendono quasi esclusivamente dalla temperatura);
3. la superficie dello scambiatore naturalmente rimane invariata.

Con queste ipotesi si può stimare i nuovi valori di A e B :

$$A' = \frac{SU}{m'_C c_{p,C}} = \frac{Am_C}{m'_C} = \frac{m_C}{m'_C} \frac{T_{C,out} - T_{C,in}}{LMTD} \quad (4.15)$$

e analogamente

$$B' = \frac{m_H}{m'_H} \frac{T_{H,in} - T_{H,out}}{LMTD} \quad (4.16)$$

Utilizzando i nuovi coefficienti si può risalire alle nuove temperature in uscita mediante il sistema 4.13 nella pagina precedente. Chiaramente questa approssimazione ha validità solo nel caso di variazioni contenute delle portate (cosa che si verifica come se vedrà in seguito).

4.3 Funzione obiettivo ottenuta dalla linearizzazione

Scelto il modello dello scambiatore e le approssimazioni si ottiene quindi che il sistema da risolvere per il calcolo della funzione obiettivo è un sistema lineare. La matrice (si riporta come esempio un singolo treno) risulta di questa tipologia:

$$\begin{pmatrix}
 1 & 0 & & & -a_{1,1} & & & & & & \\
 0 & 1 & 0 & & -a_{1,2} & & & & & & \\
 & -b_{2,1} & 1 & 0 & & & & & & & \\
 & -b_{2,2} & 0 & 1 & & & & & & & \\
 & & & -b_{3,1} & 1 & 0 & -a_{3,1} & & & & \\
 & & & -b_{3,2} & 0 & 1 & -a_{3,2} & & & & \\
 & & & & & \ddots & \ddots & & \ddots & & \\
 & & & & & \ddots & & \ddots & \ddots & & \\
 & & & & & & -b_{n,1} & 1 & & & \\
 & & & & & & -b_{n,2} & & 1 & &
 \end{pmatrix}
 \begin{pmatrix}
 T_{1,H}^{out} \\
 T_{1,C}^{out} \\
 T_{2,H}^{out} \\
 T_{2,C}^{out} \\
 T_{3,H}^{out} \\
 T_{3,C}^{out} \\
 T_{4,H}^{out} \\
 T_{4,C}^{out} \\
 T_{5,H}^{out} \\
 T_{5,C}^{out}
 \end{pmatrix}
 =
 \begin{pmatrix}
 b_{1,1}T_{1,C}^{in} \\
 b_{1,2}T_{1,C}^{in} \\
 a_{2,1}T_{2,H}^{in} \\
 a_{2,2}T_{2,H}^{in} \\
 0 \\
 0 \\
 \vdots \\
 \vdots \\
 a_{n,1}T_{n,H}^{in} \\
 a_{n,2}T_{n,H}^{in}
 \end{pmatrix}$$

doven è il numero di scambiatori. Le equazioni sono a due a due associate con gli scambiatori. Le variabili come si è detto sono il doppio del numero degli scambiatori. Come si può vedere è una sparsa, tipica dei sistemi composti da sottosistemi accoppiati. Come si può vedere le temperature del fluido “freddo” sono accoppiate ai coefficienti *b* mentre quelle del fluido “caldo” con i coefficienti *a*. Tramite la numerazione degli scambiatori si può ottenere una disposizione più compatta degli elementi della matrice, ad esempio numerando in successione gli scambiatori appartenenti allo stesso treno (quindi in coerenza con gli accoppiamenti) partendo dal primo all’ultimo (come è presente nella matrice per quanto riguarda il fluido freddo). Le temperature in ingresso di confine (ovvero nelle linee degli scambiatori prive di connessioni in ingresso) compaiono nei termini noti. Se uno scambiatore non è accoppiato ad altri diventa certamente indipendente. Si dimostra facilmente che le due equazioni associate ad ogni scambiatore date le temperature in ingresso sono linearmente indipendenti. Risulta anche possibile dimostrare che i sistemi trattati, che descrivono delle reti di scambiatori che non presentino strutture paradossali(ad esempio due scambiatori completamente accoppiati tra loro senza dipendenze esterne), risultano caratterizzate da matrici associate non singolari (date delle temperature in ingresso, le temperature in usci-

ta sono determinate e univoche). Dalla risoluzione di questo sistema, per cui si è scelto di utilizzare l'algoritmo di Doolittle, si possono quindi ottenere alcune temperature utilizzate per calcolare la funzione obiettivo vera e proprio (su cui operare l'ottimizzazione):

$$E = -\left(\sum H_{i,H}(T_{i,H}^{out}, m_{i,H}) + \sum H_{j,C}(T_{j,C}^{out}, m_{j,C})\right) \quad (4.17)$$

Dove con $H_i = m_i \bar{c}_p T_i$ è l'entalpia totale. In quest'unica fase (a parte nel calcolo di determinate temperature incognite iniziali) si utilizzano le stime delle proprietà fisiche del fluido. Resta comunque di interesse calcolare tutte le temperature nell'esigenza di determinate condizioni opzionali su queste ultime.

4.4 Caratteristiche dei vincoli

Per quanto riguarda le funzioni di vincolo si è scelto, anche in accordo con le esigenze del problema di riferimento e di quelli futuri, di condensare le singole imposizioni su determinate duties in un'unica funzione vincolo.

Date quindi le linee scelte, invece che operare con dei vincoli dati da:

$$g_i = \left| Q_i^{ris}(m) - Q_i^{ob} \right| \leq \varepsilon_i \text{ con } i = 1, \dots, n_{duty}$$

dove Q_i^{ris} sono le duties calcolate in funzione delle nuove portate e Q_i^{ob} le duties volute, sempre in riferimento a determinate linee scelte. Si ottiene il singolo vincolo:

$$\begin{cases} NQ = \frac{\|Q_t^{ris}(m) - Q_t^{ob}\|_2}{\|Q_t^{ob}\|_2} \leq \varepsilon \\ Q_t^{ris} = (Q_0^{ris}, Q_1^{ris}, \dots, Q_{n_{duty}}^{ris}) \\ Q_t^{ob} = (Q_0^{ob}, Q_1^{ob}, \dots, Q_{n_{duty}}^{ob}) \end{cases} \quad (4.18)$$

In questo modo si ottiene un risultato analogo rispetto ai singoli vincoli semplificando il problema. Questa procedura è in accordo con la condizione che le duties abbiano più o meno lo stesso ordine di grandezza (non vi sia una disparità

eccessiva tra le varie duties) e si desidera uno scostamento percentuale più o meno simile per le varie duties. Infatti si rinuncia ad avere ε_i diverse scarti, agendo solamente su un singolo scarto. Il software può comunque essere implementato non difficilmente per operare con i singoli vincoli.

4.5 Strategia di risoluzione del problema vincolato

Ritornando al problema generale si espone ora la teoria su cui si basa la risoluzione. Essa è una teoria generalizzata del metodo dei moltiplicatori di Lagrange. Definiamo ora il problema:

$$x_s : \begin{cases} x_s = \min_{x \in \mathbb{R}^n} L(x, \lambda^*) & \lambda^* \in \mathbb{R}_+^m \\ L(x, \lambda) = E(x) + \sum_{j=1}^m \lambda_j g_j(x) & \text{Lagrangiana} \end{cases} \quad (4.19)$$

Dove λ^* è il vettore di Kuhn-Tacker, per cui si verifica che la soluzione di problema è equivalente a quella del problema di partenza 4.3 . Con ciò vengono rispettate le condizioni di Kuhn-Tucker e secondo il seguente teorema (all'interno vi è una condizione di regolarità dei vincoli):

Teorema *Data la seguente condizione di Slater (regolarità dei vincoli): siano $g_j : \mathbb{R}^n \rightarrow \mathbb{R}, j = 1, \dots, m$ funzioni continue convesse, $E : \mathbb{R}^n \rightarrow \mathbb{R}$ una funzione convessa continua e*

$$\exists \bar{x} : g(\bar{x}) < 0 \quad (4.20)$$

Se x_s è una soluzione di 4.3 allora esiste $\lambda^ \in \mathbb{R}_+^m$ tale che (x_s, λ^*) è un punto di sella per la Lagrangiana. Inoltre λ^* è vettore di Kuhn-Tucker per il problema 4.3 e $E(x_s) = L(x_s, \lambda^*)$. Viceversa se (x_s, λ^*) è un punto di sella per la Lagrangiana, allora x_s è una soluzione di 4.3 e λ^* è vettore di Kuhn-Tucker.*

Va detto che il precedente problema presenta delle condizioni ben definite. In particolare la convessità delle funzioni, di cui non si hanno conoscenze analitiche.

L'algoritmo utilizzato nel software si basa su una teoria che estende l'applicabilità anche a funzioni non convesse (teoria Lagrangiana generalizzata). In questo caso la funzione Lagrangiana detta generalizzata si presenta come:

$$\bar{L}(x, \lambda) : \begin{cases} E(x) + \sum_{j=1}^m \lambda_j g_j^+(x) & \text{se } \forall k : 1 \leq k \leq m, \lambda_k \geq 0 \\ -\infty & \text{se } \exists k : 1 \leq k \leq m, \lambda_k < 0 \end{cases} \quad (4.21)$$

Con $g_j^+(x) = \max(0, g_j(x))$. Da notare che $\bar{L}(x, \lambda)$ non è differenziabile ovunque anche se g_j e E risultano differenziabili, questo però non risulta un problema. Definendo:

$$\bar{E}(x) : \begin{cases} \bar{E}(x) = E(x) & \text{se } x \text{ rispetta i vincoli} \\ +\infty & \text{altrimenti} \end{cases} \quad (4.22)$$

si nota subito che $\bar{E}(x) = \max_{\lambda \in \mathbb{R}^m} \bar{L}(x, \lambda)$. Se x_s è soluzione del problema d'origine e la Lagrangiana presenta un punto di sella per la data soluzione si ha:

$$E(x_s) = \min_x \bar{E}(x) = \min_x \max_{\lambda} \bar{L}(x, \lambda) = \max_{\lambda} \min_x \bar{L}(x, \lambda) = \max_{\lambda} \bar{\varphi}(\lambda) \quad (4.23)$$

È stato dimostrato che $\bar{\varphi}(\lambda)$ è una funzione concava, quindi non presenta grandi complicazioni risolvere il problema duale di ricerca del massimo 4.23. L'algoritmo utilizzato nel software calcola un sub-gradiente approssimato della funzione $\bar{\varphi}(\lambda)$. Risulta dimostrabile che un sub-gradiente è dato dal vettore composto da $g_j^+(x)$. Una serie di teoremi che non si riportano per sintesi (10,9,8) porta anche al seguente risultato:

Teorema *La classe di problemi per cui \bar{L} presenta un punto di sella (per cui si ha la soluzione del problema originario) contiene la classe di problemi per cui la Lagrangiana ordinaria non presenta un punto di sella.*

In pratica l'algoritmo può trovare soluzioni anche in casi in cui la Lagrangiana ordinaria non presenti un punto di sella in corrispondenza della soluzione x_s . Nel paragrafo dedicato al funzionamento dell'algoritmo sarà esposta la tecnica euristica utilizzata per il calcolo numerico della soluzione.

5 Analisi parametrica per la validazione del modello

Le misure ottenute dei flussi involventi nei processi di raffineria possono presentare incertezze elevate, quindi introdurre errori nel calcolo della previsione. Si è quindi attuata un'analisi di sensibilità parametrica per verificare gli effetti di determinati errori sui parametri conosciuti, riscontrabili su una previsione per consistenti variazioni delle portate (variabili su cui si opera per ottimizzare il sistema). Per quanto riguarda le temperature, misurate con termocoppie sembrerebbe plausibile considerare abbastanza attendibili le misure. Il problema diventa più consistente nella valutazione delle portate, misurate con con misuratori a diaframma che possono risentire dello sporco dovuto alle caratteristiche del fluido. Ci si è quindi concentrati sulla valutazione degli effetti sul modello di calcolo dovuti a errori sulle portate. Successivamente si sono fatte considerazioni sull'errore causato dall'aver trascurato il parametro f (eq. 4.10 a pagina 31), dalla considerazione di calori specifici costanti e dalla variabilità del coefficiente di scambio termico globale U (verifica generale delle ipotesi sulle approssimazioni). Si è infine verificato l'errore introdotto nell'evenienza di calcolo di una determinata temperatura incognita (non misurata) mediante l'equazione della conservazione dell'energia.

Si è scelto quindi uno scambiatore di riferimento di cui si aveva la disponibilità di tutte le misure:

Temperature [°C]	
Th,in	171.51
Th,out	128.94
Th,in	86.67
Th,out	100.62
Portate [kg/h]	
MHi	22411.95
MCi	90487.65

Questi dati sono quindi stati utilizzati nel modello esposto in precedenza ovvero il calcolo di A e B con le nuove portate e l'ottenimento delle nuove temperature di uscita. I casi analizzati si riferiscono a variazioni consistenti delle portate, per comprenderne l'entità sono riportati i coefficienti moltiplicativi rispetto al dato iniziale misurato. L'errore sulle temperature in uscita è stato calcolato mediante:

$$err_{\%} = 100 \frac{|T_{err} - T_{cor}|}{T_{cor}}$$

5.1 Analisi delle conseguenze di eventuali errori sulle portate misurate

Si è quindi proceduto al calcolo delle temperature in uscita con nuove portate stabilite (aumento o diminuzione), in relazione a eventuali valori di errore nelle portate misurate (parametri iniziali). Si è agito applicando prima l'errore a una portata iniziale (parametro), poi all'altra e infine a entrambe, con una riduzione o maggiorazione contemporanea (casi di impatto maggiore sui coefficienti del modello). Gli errori sulle portate sono stati imposti mediante un grado di maggiorazione-riduzione $M_{err} = M_i * grd$ (con ad es. $grd = 0.8, 0.9, 1.1, 1.2$). Le temperature senza errori sono riportate per migliorare la visualizzazione del confronto.

Con variazione di portate verso valori maggiori rispetto a quelle iniziali:

MC	180975.3	2*MCi
MH	44823.9	2*MHi

si ottengono le seguenti nuove temperature in uscita in relazione all'errore della stima della portata calda iniziale (parametro):

MH (valore iniziale moltiplicato per grd)						
Th	137.7	137.829	137.957	138.212	138.338	138.464
Tc	91.1005	91.7141	92.3231	93.5269	94.1218	94.7122
Th corr	138.085	138.085	138.085	138.085	138.085	138.085
Tc corr	92.9273	92.9273	92.9273	92.9273	92.9273	92.9273
grd	0.7	0.8	0.9	1.1	1.2	1.3

e i relativi errori

err % Th	0.278372	0.185168	0.092377	0.091966	0.183521	0.274668
err % Tc	1.96586	1.3055	0.650227	0.645214	1.28545	1.92074
grd	0.7	0.8	0.9	1.1	1.2	1.3

Per la portata fredda:

MC						
Th	146.409	143.493	140.72	135.579	133.197	130.933
Tc	93.3828	93.2261	93.0743	92.7849	92.6469	92.5132
Th corr	138.085	138.085	138.085	138.085	138.085	138.085
Tc corr	92.9273	92.9273	92.9273	92.9273	92.9273	92.9273
grd	0.7	0.8	0.9	1.1	1.2	1.3

e i relativi errori

err % Th	6.02858	3.91664	1.9088	1.81459	3.53955	5.17925
err % Tc	0.490125	0.321515	0.1582	0.153263	0.301758	0.44565
grd	0.7	0.8	0.9	1.1	1.2	1.3

Con tutte e due le portate affette da errore:

MC e MH						
Th	146.099	143.268	140.6	135.712	133.475	131.363
Tc	91.427	91.9569	92.4564	93.3714	93.7903	94.1856
Th corr	138.085	138.085	138.085	138.085	138.085	138.085
Tc corr	92.9273	92.9273	92.9273	92.9273	92.9273	92.9273
grd	0.7	0.8	0.9	1.1	1.2	1.3

e i relativi errori

err % Th	5.8041	3.75395	1.8217	1.71797	3.33858	4.86772
err % Tc	1.61453	1.04424	0.506744	0.477891	0.928695	1.35406
grd	0.7	0.8	0.9	1.1	1.2	1.3

Passando ora al caso di diminuzione delle portate (sulle variabili):

MC	45243.825	0.5*MCi
MH	11205.975	0.5*MHi

Per portata calda affetta da errore:

MH						
Th	101.206	101.793	102.376	103.531	104.103	104.671
Tc	95.8828	97.111	98.3178	100.668	101.812	102.936
Th corr	102.956	102.956	102.956	102.956	102.956	102.956
Tc corr	99.5035	99.5035	99.5035	99.5035	99.5035	99.5035
grd	0.7	0.8	0.9	1.1	1.2	1.3

e i relativi errori

err % Th	1.69993	1.12964	0.562976	0.559222	1.11462	1.66614
err % Tc	3.63877	2.40446	1.19161	1.1706	2.32043	3.44972
grd	0.7	0.8	0.9	1.1	1.2	1.3

MC						
Th	112.547	108.747	105.586	100.765	98.9386	97.4133
Tc	102.439	101.357	100.382	98.7096	97.9913	97.34
Th corr	102.956	102.956	102.956	102.956	102.956	102.956
Tc corr	99.5035	99.5035	99.5035	99.5035	99.5035	99.5035
grd	0.7	0.8	0.9	1.1	1.2	1.3

err % Th	9.31564	5.62475	2.55475	2.12769	3.90179	5.38327
err % Tc	2.94978	1.86251	0.883229	0.797841	1.51979	2.17426
grd	0.7	0.8	0.9	1.1	1.2	1.3

MC e MH						
Th	110.675	107.52	104.989	101.321	100.006	98.9476
Tc	98.0585	98.6491	99.1229	99.8095	100.056	100.254
Th corr	102.956	102.956	102.956	102.956	102.956	102.956
Tc corr	99.5035	99.5035	99.5035	99.5035	99.5035	99.5035
grd	0.7	0.8	0.9	1.1	1.2	1.3

err % Th	7.49723	4.43292	1.97456	1.58749	2.86476	3.89307
err % Tc	1.45219	0.858642	0.382467	0.307491	0.554895	0.754076
grd	0.7	0.8	0.9	1.1	1.2	1.3

Il primo risultato da notare è che, maggiore è la variazione di portata, maggiore è l'impatto dell'errore a parità di errore percentuale iniziale. Si noterà anche che l'errore sulle temperature è evidentemente maggiore in quelle relative alla portata affetta da errore iniziale. L'entità degli errori non è comunque drammatica. Si deve considerare anche che è ragionevole pensare che il misuratore di portata commetta più o meno uniformemente l'errore registrato sulle misure, ovvero che nel caso di modifica delle portate in range non troppo elevati si registri un comportamento analogo e un errore simile presente nell'imposizione della nuova portata. L'errore delle portate comunque agisce due volte e risulta notevolmente critico,

esso si ripercuote non solo sui parametri del modello e quindi sul calcolo delle nuove temperature, ma anche sulle nuove portate da elaborare con conseguenti errori ulteriori sul risultato (non solo si trova un punto di massimo errato, ma anche se si fosse trovato giusto si andranno ad applicare portate diverse). Si può notare come le condizioni con errore maggiore si verifichino in corrispondenza dell'operazione di riduzione delle portate, in particolare della portata fredda (di maggiore entità). Da sottolineare anche il fatto che errori di sottostima o sovrastima di entrambe le portate possano compensare gli errori generali, dando risultati migliori rispetto alla singola portata perturbata.

5.2 Considerazioni sugli altri parametri affetti da approssimazioni

Anche su questi parametri sono state fatte analisi simili a quella precedente, con le stesse variazioni delle variabili portate. Sono riportate in [riferimento]. Per quanto riguarda il coefficiente di scambio termico globale gli errori risultano contenuti, e crescono anch'essi al crescere della variazione delle condizioni di regime (con non eccessive variazioni di portate risultano trascurabili). Per quanto riguarda i calori specifici e il coefficiente di correzione, gli effetti risultano veramente contenuti e quindi ragionevolmente trascurabili. In appendice A si trattano anche le eventuali perdite di calore verso l'esterno. Anch'esse risultano trascurabili. Diverso è il caso di una temperatura in uscita incognita, che viene calcolata mediante la legge di conservazione dell'energia approssimata. In questo caso infatti compaiono nel calcolo i valori assoluti delle portate, e non più i rapporti come nei coefficienti A e B , ne risulta quindi che sia opportuno inserire il più possibile tutti i dati delle temperature delle misurazioni. Nell'uso del software è quindi consigliato non richiedere il calcolo di più di 2-3 temperature.

5.3 Conclusioni dell'analisi parametrica

Dall'analisi parametrica è risultato come le approssimazioni imposte siano del tutto ragionevoli e non prive di fondamento. Va detto che l'analisi parametrica è stata fatta su un singolo scambiatore e non sull'intero sistema. Gli scambiatori essendo connessi tra di loro possono portare a una propagazione degli errori che vada a inficiare i risultati. Una forte ipotesi che deve essere verificata per il buon funzionamento del software è la misura precisa delle temperature. Infatti la verifica del buon esito dell'ottimizzazione si basa necessariamente sulla misura delle temperature in uscita. Nel caso in cui sia avvenuta comunque l'ottimizzazione e si sia raggiunto un miglioramento, questo potrebbe essere non visibile a causa degli errori nella misura delle temperature. D'altra parte non è possibile in nessun modo sopperire a mancanze di informazioni eccessive in maniera teorica. Nell'applicazione del software quindi va valutata l'opportunità di calibrare in modo preciso i misuratori, o di attuare dei miglioramenti di questi ultimi.

6 Architettura e implementazione del software

Il software, come è stato detto in precedenza è scritto nel linguaggio C++. I motivi per cui si è scelto di utilizzare questo linguaggio consistono nella possibilità di sfruttare la sua versatilità e velocità unitamente alla non eccessivamente complicata procedura di interfaccia con il programma operante sul server e che accede alle misure di campo. L'interfaccia viene realizzata grazie alla creazione di una libreria eseguibile di tipo ATL (*Active Template Library*) ovvero corredata da una serie di classi basate su C++ che rendano agile la creazione di applicazioni di un certo livello in particolare oggetti COM (interfaccia COM - *Component Object Model*) che prevedano un'interfaccia con qualsiasi linguaggio di programmazione che supporti questa tecnologia (in questo caso il software del server). In sostanza il codice che calcola l'ottimizzazione è completamente scritto in C++ , per la precisione è una classe nominata *optimizer* (file: *optimizer.h*) che provvede a procurare lo spazio di memoria utilizzato per il calcolo, alla registrazione dei da-

ti, delle variabili intermedie e dei risultati, all'esecuzione di varie funzioni e alla creazione di sotto-classi. Si sottolinea che

- ogni riga di codice del programma è conosciuta nelle proprie finalità (si ha la completa conoscenza del suo ruolo);
- il codice è stato quasi interamente scritto dall'autore di questa tesi se non per un *template* preesistente utilizzato per la risoluzione di sistemi lineari che è stato solo in parte modificato (LUfactor.h).

Nei prossimi paragrafi saranno esposti la struttura del software e le caratteristiche degli algoritmi utilizzati (metodi numerici, particolari accorgimenti ecc.).

6.1 Struttura complessiva

La libreria completa è composta da un'innumerabile serie di file, tra cui dipendenze esterne, che non saranno descritti (totalmente inutile in ambito di questo lavoro). Lasciando da parte i template che realizzano l'interfaccia si prenderà ora in considerazione solamente la classe ottimizzatrice optimizer.h (l'unico file direttamente richiamato dall'ATL) e i file secondari che utilizza. I file che compongono il codice atto all'ottimizzazione sono sette:

1. optimizer.h
2. structure.h
3. fluid.h
4. LUfactor.h
5. NelderMead.h
6. NelderMeadCs.h
7. vector_opr.h

Alcune di questi file sono template, nell'intenzione di renderli versatili in determinati utilizzi su classi diverse (ad esempio `vector_opr` aggiunge delle operazioni sui vettori che potranno essere utilizzate con numeri interi, numeri in virgola mobile ecc.). Tutti questi file sono file di intestazione (obbligatoriamente per i template), questa è una scelta personale dell'autore ed è dettata da motivi di comodità e comprensibilità. Il programma è stato concepito in modo tale da avere una discreta velocità di esecuzione e la massima portabilità. Per ottenere questo si è scelto di non utilizzare librerie precostituite, di cui non si conosce esattamente il funzionamento e possono presentare complessità non richieste. Sono utilizzati alcuni template della libreria standard di C++, precisamente:

- “vector”, template per la definizione e le operazioni sui vettori in senso matematico);
- “cmath”, template per l'utilizzo di determinate funzioni matematiche come il logaritmo o l'esponenziale;
- “limits”, per l'utilizzo del concetto matematico di infinito;
- “ctime”, per l'accesso all'orario allo scopo di generare numeri pseudocasuali.

6.1.1 Descrizione sintetica dei singoli file

optimizer.h Questo file contiene la definizione di una classe omonima (`optimizer`) che contiene la procedura per l'ottimizzazione. Tramite essa si ha la registrazione dei dati utili, delle variabili intermedie e dei risultati. In essa si creano le sottoclassi che ricoprono il ruolo di determinate strutture o che eseguono determinate funzioni.

structure.h Contiene la definizione di una classe omonima (`structure`) che consiste in una struttura virtuale che riproduce le entità scambiatori o split nel modello

matematico. La scelta di definire questa struttura è dettata dalla necessità di riprodurre in modo ordinato e comprensibile il modello matematico in ambito virtuale. Ogni informazione riguardante ad esempio uno scambiatore è memorizzata in un oggetto “structure”. Durante il calcolo varie funzioni contenute in optimizer.h accedono a queste sotto-strutture.

fluid.h Contiene la definizione di una classe omonima che consiste in una struttura virtuale che memorizza le caratteristiche di un determinato fluido involvente nel sistema di scambiatori; archivia le proprietà del fluido (calore specifico, densità) e le funzioni per il calcolo in altre condizioni termodinamiche e a uno scopo analogo alla classe precedente.

LUfactor.h: In questo file è definita la classe template denominata LU, si tratta di un file precostituito e proviene da una parte (JAMA - una serie di template per l'algebra lineare) del pacchetto TNT (Template Numerical Toolkit¹) elaborato dal NIST (National Institute of Standards and Technology). La classe è stata studiata approfonditamente e adattata all'uso del template “vector” invece che le strutture comprese nel Toolkit. In particolare utilizza l'algoritmo di Doolittle con pivotig parziale (righe).

NelderMead.h Intestazione della classe template che ha il ruolo di una funzione e ricerca il minimo globale di un campo scalare restituendo il punto di massimo e il valore della funzione in tal punto. L'algoritmo su cui si basa il metodo di ricerca è il metodo Nelder-Mead.

NelderMeadCs.h Come il file precedente ma con la classe adattata al caso di ottimizzazione vincolata. Utilizza lo stesso algoritmo di ricerca di NelderMead.h ma con delle modifiche basate sulla teoria della Lagrangiana generalizzata per il rispetto dei vincoli imposti.

¹<http://math.nist.gov/tnt/>

vector_opr.h Questo file contiene alcune funzioni template che consistono in operazioni tra vettori aggiuntive rispetto a quelle di base incluse in “vector”, ad esempio la norma vettoriale, o il prodotto scalare, o la somma tra vettori.

6.1.2 Schema generale della classe ottimizzatrice

Il prospetto informativo espone tutte le funzioni di interfaccia per l'utilizzo della libreria. In questo paragrafo ci si appresta a descrivere in maniera più dettagliata la modalità di funzionamento della libreria:

1. appena la libreria viene eseguita provvede a creare un oggetto (variabile globale) di tipo optimizer che conterrà tutto ciò che è necessario per l'ottimizzazione; in questo oggetto saranno inseriti i dati e la configurazione del problema. Mediante la funzione initialize viene avviata la procedura di calcolo e quindi saranno disponibili i risultati. Nella funzione initialize il programma provvede anche a calcolare alcune temperature in uscita dagli scambiatori eventualmente non inserite. Il calcolo utilizza la prima equazione del sistema 4.9 a pagina 31, utilizzando le densità e i calori specifici stimati. Essendo una sola equazione risulta possibile calcolare solo una temperatura per scambiatore.
2. All'interno dell'oggetto optimizer vengono creati degli altri oggetti di altre classi e riempiti con i dati; il sistema provvede a completare in automatico le strutture virtuali su cui si opera il calcolo. Questi oggetti si possono dividere in due tipologie principali:
 - (a) gli oggetti che riproducono virtualmente il problema fisico;
 - (b) gli strumenti che servono alla risoluzione del problema e alla ricerca operativa.
3. Tutti i sotto-oggetti si riferiscono all'oggetto principale di tipo optimizer e interagendo con esso riescono ad accedere alle diverse funzioni vicendevolmente.

4. Gli oggetti che riproducono il sistema fisico consistono in:
 - (a) structure, unità virtuali che riproducono le componenti della rete di scambiatori (split, scambiatori) e forniscono le equazioni associate per la risoluzione;
 - (b) fluid, oggetti che contengono le informazioni e le funzioni per il calcolo di determinate proprietà dei fluidi in un determinato stato termodinamico imposto.

5. Gli oggetti che garantiscono la risoluzione del problema consistono in:
 - (a) LU, che esplica la risoluzione del sistema lineare per il calcolo della funzione obiettivo;
 - (b) NelderMead e NelderMeadCs, che provvedono alla ricerca operativa con o senza vincoli imposti (diseguazioni);

La scelta di questo schema logico è dettata dal fatto di avere un ordine comprensibile e una linearità nello schema di risoluzione del problema. Le strutture si prestano in modo ottimale alla riproduzione del problema fisico. Le sottoclassi per la risoluzione permettono di essere indipendenti dal sistema ed essere utilizzate anche in altri ambiti con piccole modifiche, quindi consistono in veri e propri tool. Tutta la procedura è eseguita all'interno della classe optimizer. Quando distrutta la classe libera tutta la memoria utilizzata per il calcolo e la memorizzazione dei dati, occupata al momento della creazione. La libreria permette quindi di avere un oggetto completamente indipendente dal sistema che lo esegue. Quest'ultimo si interfaccia con esso tramite delle funzioni e non può in nessun modo modificare ciò a cui non gli è consentito l'accesso. Le funzioni di interfaccia prevedono anche l'impostazione dello schema della rete di scambiatori. Ciò permette quindi una completa versatilità nel caso di modifiche dello schema nel tempo, senza la necessità di modifiche del codice e di ricompilazioni. La classe optimizer provvede a ricevere i dati in modo ordinato, senza dover inserire informazioni ridondanti. Infatti procede al completamento del sistema in base alle connessioni in

ingresso delle varie strutture fornite dalle funzioni di interfaccia. Si ottiene quindi una robustezza nel funzionamento, per evitare il più possibile effetti indesiderati. Le informazioni in uscita comprendono anche segnalatori che comunicano l'esito della procedura. Quindi ad esempio se si verifica un inserimento di dati insufficiente esso sarà immediatamente visibile al termine dell'esecuzione. Come è stato sottolineato in precedenza il software necessita di una buona completezza dei dati di campo, in particolare:

- è strettamente necessaria la conoscenza delle temperature in ingresso al sistema complessivo;
- è strettamente necessaria la conoscenza delle portate in ingresso;
- è consigliato di inserire il maggior numero possibile di misurazioni delle temperature in uscita da ogni scambiatore e necessariamente almeno una per ogni scambiatore.

Tra le varie possibilità vi è quella di poter impostare un range di scostamento sulle portate iniziali (prima del calcolo dell'ottimizzazione) che si richiede di rispettare; questa opzione è in relazione alle esigenze dell'applicazione, come si vedrà in seguito. Oltre a ciò è possibile monitorare lo sporco degli scambiatori mediante l'accesso al parametro $A \cdot U$ (area per coefficiente di scambio termico globale), con cui valutare i fenomeni di sporco e le opportunità di mettere in manutenzione determinate strutture.

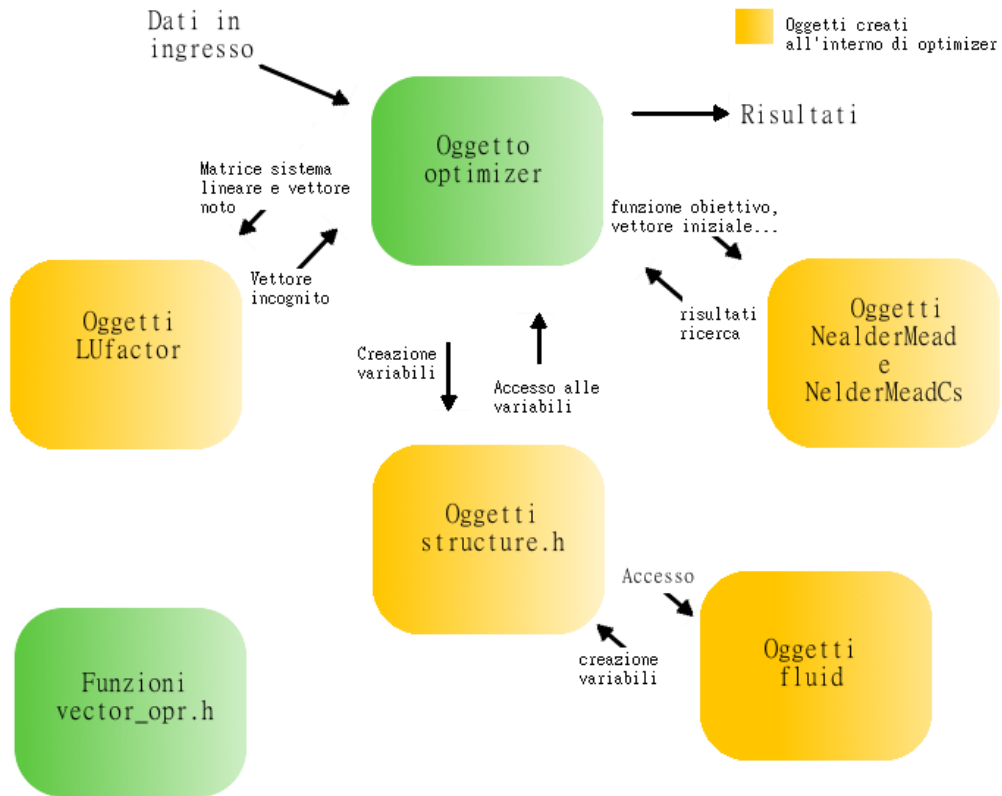


Figura 6.1: schema di funzionamento dell'oggetto di tipo optimizer

6.1.3 Esecuzione della libreria e differenti approcci in funzione del tipo di ottimizzazione

Dopo l'inserimento di tutti i dati necessari mediante le funzioni di interfaccia il programma viene inizializzato al calcolo mediante la funzione initialize. In essa sono presenti delle funzioni interne che vengono eseguite e passo passo elaborano il problema fino alla risoluzione. Quindi in ordine vengono inizialmente completate le strutture per la schematizzazione del sistema di scambiatori e a seconda delle modalità di ottimizzazione (vincolata o libera) vengono eseguite le classi ottimizzatrici. Nel caso di ottimizzazione libera si otterrà il nuovo vettore

di portate che massimizza la funzione obiettivo inserita. Nel caso di ottimizzazione vincolata il software opererà in primis per verificare se esiste un vettore di gradi di frazionamento degli split che consente il rispetto dei vincoli, ciò avviene minimizzando lo scarto derivante da 4.18 a pagina 35:

$$SQ = \frac{\|Q_t^{ris}(m) - Q_t^{ob}\|_2}{\|Q_t^{ob}\|_2} - \varepsilon \quad (6.1)$$

Quindi se l'algoritmo per la minimizzazione libera riesce a trovare un valore dello scarto minore di 0 si procede all'ottimizzazione vincolata, al contrario se questo non avviene verrà accettato il vettore di portate che minimizza tale scarto (l'ottimizzazione viene messa da parte e si propone l'obiettivo di raggiungere il più possibile le duties volute). Nel caso di possibilità di ottimizzazione vincolata il programma procede alla creazione dell'oggetto di classe NelderMeadCs e a trovare il vettore di portate che oltre a rispettare i vincoli massimizza la funzione obiettivo. Tra i risultati sarà possibile naturalmente verificare NQ e quindi il rispetto dei vincoli o lo scostamento da essi. In seguito saranno descritti i metodi numerici in cui si traducono i modelli matematici e le strategie di risoluzione.

6.2 Modelli numerici e implementazione del codice

In precedenza è stato esposto il modello matematico che definisce il problema. La risoluzione del problema permette di ottenere il risultato cercato. Il problema viene risolto numericamente, è necessario quindi descrivere le modalità con cui il modello matematico viene tradotto a livello numerico. I modelli numerici e le implementazioni principali riguardano le funzioni dell'unità scambiatore di calore, l'algoritmo di risoluzione del sistema lineare associato alla rete composta dalle varie unità e gli algoritmi di ricerca del minimo.

6.2.1 Implementazione delle funzioni associate all'unità scambiatore di calore

Riprendendo il modello matematico dello scambiatore nella sezione 4.2.1 a pagina 29 si espone ora la modalità di implementazione. La traduzione non presenta grandi difficoltà, si tratta semplicemente di ottenere le funzioni dei coefficienti delle equazioni (che poi andranno a comporre la matrice del sistema complessivo) in ambito numerico. Si nota come le varie funzioni sono prolungate per continuità in $A - B = 0$ (dove nella funzione utilizzata per $A - B \neq 0$ si avrebbe una divisione per 0), risultando anche derivabili ovunque. Numericamente il prolungamento delle funzioni è stato realizzato assegnando il valore limite per valori di $|A - B| < \varepsilon$, dove ε è un numero sufficientemente piccolo ($1e-6$). Infatti assumendo semplicemente la condizione $A - B = 0$ si rischia per valori molto piccoli di $A - B$ di ottenere dei valori errati della funzione a causa degli errori di arrotondamento nelle varie operazioni. Le funzioni al di fuori di $A - B \cong 0$ sono composte da esponenziali. Da un semplice studio analitico si può verificare che per valori estremi del campo reale tendono a dei limiti precisi (0 e 1), ciò può essere osservato anche nei grafici riportati. A e B assumono solamente valori maggiori di zero e poiché sono inversamente proporzionali ognuno a una delle due portate fluenti nello scambiatore, possono presentare valori grandi per portate piccole. Poiché gli esponenziali sono in funzione di $A - B$ o $B - A$ si può verificare che se una delle due portate assume un valore troppo piccolo il valore degli esponenziali può risultare maggiore del massimo numero rappresentabile; in precisione doppia (64 bit):

$$MAX \cong 1.80 \cdot 10^{308} \quad \ln(MAX) \cong 709$$

Quindi per valori di $|A - B|$ maggiori di 600 si è provveduto a sostituire il valore delle funzioni con gli appropriati valori dei limiti. Ciò è possibile in ragione del fatto che i valori delle funzioni possono essere approssimati ai limiti per contenuti valori assoluti di $A - B$, ($|A - B| > 5$), e tali valori decrescono con l'aumentare

dei valori assoluti di B e A . Le funzioni dei coefficienti sono membri degli oggetti di classe structure e accedono direttamente alle variabili intermedie contenute in queste ultime ($A \in B$, all'evenienza ricalcolati tramite funzioni membro sulle nuove portate inserite nelle structure).

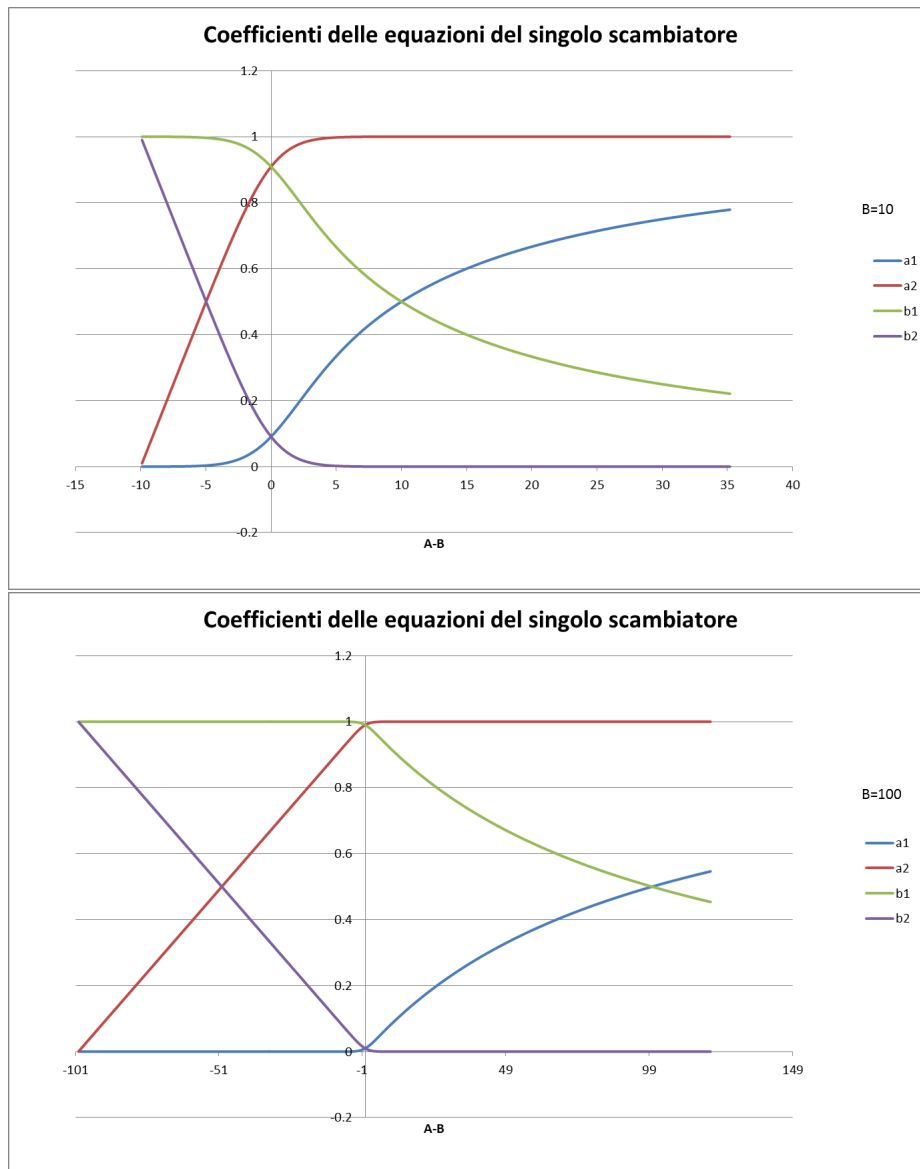


Figura 6.2: grafici delle funzioni a_1, a_2, b_1, b_2 in funzione di $A - B$ per due valori di B

6.2.2 Algoritmo per la risoluzione di sistemi lineari e caratteristiche della fattorizzazione

La risoluzione dei sistemi lineari avviene tramite la classe LU. Come si è stato già accennato questa è l'unica componente del software preconstituita, ma modificata e adattata all'utilizzo nel software. Gli oggetti di questa classe ricevono come argomento una matrice (anche non quadrata) e da essi si possono ottenere:

- la fattorizzazione LU consistente nelle matrici L (matrice triangolare inferiore) e U (matrice triangolare superiore), la matrice L presenta la diagonale con termini uguali all'unità (nel caso di matrice quadrata di partenza);
- il determinante della matrice di partenza;
- la soluzione del sistema in base a un vettore di termini noti introdotto (nel caso di matrice quadrata).

Il metodo di risoluzione consiste nell'algoritmo di Doolittle con pivoting parziale (righe). La procedura è riportata in seguito. In realtà il codice presenta una procedura di pivoting parziale sulle righe ogni qual volta si calcolano i nuovi elementi u_{ij}, l_{ji} . Mediante la tecnica di pivoting la fattorizzazione esiste sempre anche se la matrice è singolare. Per la natura dei sistemi analizzati non si dovrebbe verificare il mal condizionamento della matrice di partenza. Si è provveduto a verificare l'esattezza della soluzione in vari test, ricalcolando il termine noto dalla soluzione e la matrice originaria. Risulta possibile mantenere questo accorgimento come elemento fisso del codice anche se introduce operazioni aggiuntive (maggiore tempo di calcolo).

Va sottolineato che sebbene l'algoritmo di fattorizzazione non sia probabilmente il migliore per la risoluzione di sistemi con matrici sparse, esso presenta una discreta velocità di esecuzione oltre che una semplicità notevole. Inoltre i sistemi analizzati non presentano un numero di variabili troppo grande (<100), quindi l'utilizzo di algoritmi dedicati alle matrici sparse non dovrebbe portare a vantaggi sostanziali. Le soluzioni ottenute dalla risoluzione del sistema di scambiatori

sono state confrontate con altri software di simulazione di processo ed è risultata un'ottima congruenza (validazione del modello fisico-matematico e numerico).

Metodo di fattorizzazione di Doolittle

Data la matrice:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} = LU = \begin{bmatrix} 1 & 0 & \dots & 0 \\ l_{21} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & u_{nn} \end{bmatrix}$$

Dal prodotto matriciale si ha

$$a_{1j} = u_{1j}, \quad j = 1, 2, \dots, n$$

$$a_{ij} = \begin{cases} \sum_{t=1}^j l_{it}u_{tj}, & \text{con } j < i \\ \sum_{t=1}^{i-1} l_{it}u_{tj} + u_{ij} & \text{con } j \geq i \end{cases}$$

Si procede quindi attuando:

$$u_{1j} = a_{1j}, \quad j = 1, 2, \dots, n \text{ (prima riga di U)}$$

$$l_{j1} = a_{j1}/u_{11}, \quad j = 1, 2, \dots, n \text{ (prima colonna di L)}$$

Per $i = 2, 3, \dots, n - 1$:

$$u_{ii} = a_{ii} - \sum_{t=1}^{i-1} l_{it}u_{tj}$$

$$u_{ij} = a_{ij} - \sum_{t=1}^{i-1} l_{it}u_{tj} \quad \text{per } j = i + 1, \dots, n \text{ (j-esima riga di U)}$$

$$l_{ji} = \frac{a_{ji} - \sum_{t=1}^{i-1} l_{jt}u_{ti}}{u_{ii}} \quad \text{per } j = i + 1, \dots, n \text{ (j-esima colonna di L)}$$

Fine

$$u_{nn} = a_{nn} - \sum_{t=1}^{n-1} l_{nt}u_{tn}$$

6.2.3 Algoritmo e implementazione per la ricerca non vincolata del minimo globale

La ricerca non vincolata del minimo globale è attuata tramite la classe Nelder-Mead, che è stata completamente realizzata dall'autore di questa tesi. L'algoritmo di base è di tipo euristico e non prevede l'utilizzo di derivate. Questo porta a vari

vantaggi, come la maggior semplicità di implementazione e la possibilità di utilizzo di questo strumento anche con funzioni non regolari. Nella procedura di ricerca non vi è quindi la necessità di verificare particolari condizioni di convergenza (sebbene paradossalmente sia molto difficile dimostrare che l'algoritmo porti con certezza a convergenza). L'algoritmo funziona discretamente per un numero di variabili non eccessivo (<10). Le varie soluzioni per il miglioramento dell'algoritmo sono state testate con varie prove, in particolare utilizzando la funzione di Rosenbrock. Quest'ultima funzione, non convessa, presenta caratteristiche particolari per testare le prestazioni degli algoritmi. L'algoritmo prevede la ricerca in un determinato range delle variabili $\{x_{i,max}, x_{i,min}\}$, quindi in un sottoinsieme voluto dello spazio complessivo che risulta un iper-parallelepipedo. L'algoritmo riceve una determinata precisione scelta per il calcolo e opera in relazione ad essa e un numero massimo di valutazioni della funzione. Risulta possibile anche impostare il numero di ulteriori attuazioni dell'algoritmo da un punto iniziale generato casualmente, in modo da scongiurare l'evenienza di ottenere un minimo locale invece che globale.

Algoritmo di Nelder-Mead implementato L'algoritmo procede con le seguenti fasi:

- costruzione del semplice non degenero iniziale con uno dei vertici dato dal punto iniziale;
- fino al raggiungimento delle condizioni di terminazione si a un ciclo con la ripetizione di:
 - trasformazioni del semplice;
 - calcolo e verifica delle condizioni di terminazione;
- registrazione del vettore che presenta le migliori condizioni (valore minimo della funzione associata) e ripetizione della precedente procedura partendo

da questo vettore ma con un semplice più piccolo fino all'ottenimento di due vettori di partenza vicini;

- ripetizione per un numero determinato di volte della procedura finora esposta con un punto iniziale casuale (si vuole trovare il minimo globale);
- restituzione del vettore migliore.

Costruzione del semplice iniziale L'algoritmo costruisce inizialmente un semplice non degenere dato da $n + 1$ vettori nello spazio del vettore che ha come componenti n variabili indipendenti. Il semplice è composto naturalmente da punti presenti all'interno del sottoinsieme di ricerca scelto. Il semplice può essere di vario genere: con vertici generati casualmente o di forme determinate (semplice standard espanso, semplice regolare). Si sono testati i vari tipi di semplice. Il semplice standard espanso sembra dare buoni risultati (non vi è un evidente vantaggio dall'uso del semplice regolare che prevedere una maggiore complicazione nella costruzione). Il semplice casuale può risultare invece degenerare in alcuni casi o essere troppo allungato in alcune dimensioni rispetto ad altre (può risultare lento nel coprire alcune dimensioni durante le trasformazioni). Il semplice standard espanso viene generato dal punto iniziale fornito all'algoritmo semplicemente variando di una stessa quantità una delle componenti per $glin$ vettori da generare. Questa quantità è tale da essere almeno poco più piccola della metà del range della variabile specifica (in modo da avere un semplice abbastanza grande). I semplici di verifica, ovvero semplici più piccoli generati da un vettore ottenuto già dall'applicazione del ciclo di trasformazioni, presentano una variazione del punto iniziale circa 100 volte più piccola. La strategia di alternare semplici grandi a semplici piccoli serve a verificare che il primo tentativo con il semplice grande abbia generato un vettore realmente nei pressi di un punto stazionario.

Procedura di trasformazione del semplice La procedura di trasformazione verrà esposta per due variabili. La trattazione può essere naturalmente estesa

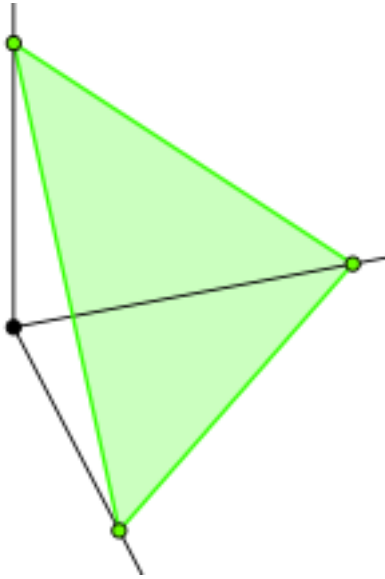


Figura 6.3: simpleso standard in tre dimensioni

a un numero indefinito di variabili. L'algoritmo prevede l'utilizzo di determinati parametri (il cui valore è consigliato dalla letteratura). Inizialmente si calcolano i valori della funzione $f(x)$ nei vari punti del simpleso e si registra quale presenta il valore maggiore, quale il secondo maggiore e quale il valore minimo: x_h, x_s, x_l a cui corrispondono f_h, f_s, f_l . Lo scopo della trasformazione è di sostituire x_h con un punto che presenti un valore della funzione minore. Quindi viene calcolato il baricentro $c = \frac{\sum_i x_i}{n}$ con tutti i punti ad esclusione del punto x_h . Dalla figura 6.4 si può vedere come la trasformazione possa avvenire in cinque modalità. I parametri scelti assumono i valori $\alpha = 1, \beta = \frac{1}{2}, \gamma = 2, \delta = \frac{1}{2}$.

L'algoritmo procede con la seguente iterazione (composta da uno o più passi):

1. riflessione ($x_r = x_h + \rho(c - x_h)$, $f_r = f(x_r)$), se $f_l \leq f_r < f_s$ accettato x_r ;
2. se $f_r < f_l$ espansione ($x_e = x_h + \gamma(c - x_h)$, $f_e = f(x_e)$), se $f_e < f_r$ accettato x_e viceversa si accetta x_r ;
3. se $f_s \leq f_r < f_h$ contrazione esterna ($x_c = c + \beta(x_r - c)$, $f_c = f(x_c)$), se $f_c \leq f_r$ si accetta x_c ;

4. se $f_r \geq f_h$ contrazione interna ($x_c = c + \beta(x_h - c)$, $f_c = f(x_c)$), se $f_c < f_h$ si accetta x_c ;
5. se non vi è stata nessuna accettazione nei precedenti passi si attua una contrazione totale:
si ottengono nuovi n vertici $x'_j = x_l + \delta(x_j - x_l)$ con $j = 1, \dots, n$, con $j \neq l$.

Questo metodo può portare a sostituire i vertici originari con vertici al di fuori dell'iper-parallelepipedo di ricerca. I passi che possono portare a questo sono le riflessioni e le espansioni. Questo problema è stato risolto inserendo dei passi aggiuntivi denominati di riduzione che riportino i punti all'interno della frontiera. Questa operazione è abbastanza delicata poiché si deve lasciare la possibilità al semplice di potersi schiacciare liberamente sulla frontiera, se su quest'ultima si trova il punto di minimo. Prendendo ad esempio in considerazione x_r (discorso analogo viene fatto per x_e) per ogni componente che si trova al di fuori del campo di ricerca si calcola:

$$k_i^{min} = \left\lceil \ln\left(\frac{x_{r,i} - c_i}{x_{max,i} - c_i}\right) \frac{1}{\ln 2} \right\rceil, k_i^{max} = \left\lceil \ln\left(\frac{x_{r,i} - c_i}{x_{min,i} - c_i}\right) \frac{1}{\ln 2} \right\rceil$$

viene quindi scelto il minimo $\bar{k} = \{k_i^{min}, k_i^{max}\}$ e utilizzato per ricalcolare un nuovo $x_r = c + (\frac{1}{2})^{\bar{k}}(x_r - c)$ che esplica una riflessione diciamo "contenuta" nelle frontiere. Naturalmente se il punto riflesso è stato ridotto non si procede con l'espansione. Il metodo trovato sembra risultare molto efficiente e permette di ottenere risultati (punto di minimo) anche sulla frontiera del campo di ricerca.

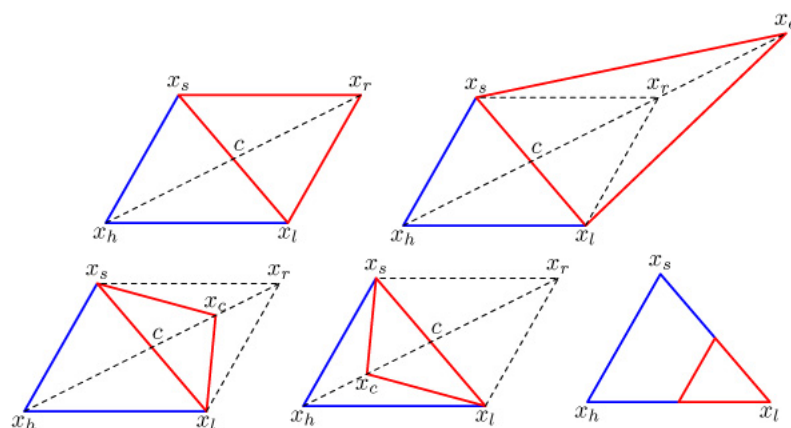


Figura 6.4: procedure di trasformazione; riflessione, espansione, contrazione interna, contrazione esterna, contrazione totale.

Calcolo delle condizioni di terminazione Vi sono innumerevoli modi per terminare le iterazioni di trasformazione del semplice. Ad esempio tramite norme sulla funzione calcolata nel punto sostituito e quello nuovo, norme sulla distanza tra questi due punti oppure stime della derivata. Oppure la somma degli scarti quadratici medi della distanza tra i punti del semplice e il baricentro (verificare se il baricentro è collassato). La condizione utilizzata nell'implementazione è invece più particolare. Questo algoritmo ha il difetto di continuare le iterazioni fino alle condizioni di terminazione, anche se il semplice è collassato e quindi si è persa una dimensione rispetto a quelle del dominio. La ricerca in questo caso procede in un sottospazio vettoriale. Ciò risulta chiaramente inefficiente, oltre a non poter condurre al punto stazionario in tale ciclo di iterazioni (finché non si genera un altro semplice non degenero). Si è trovata quindi una soluzione che possa integrare sia l'esigenza di terminare l'algoritmo quando il semplice collassa in un singolo punto (condizione che si verifica alle vicinanze di un punto di minimo anche non stazionario), sia di interrompere e poter ripartire con un nuovo semplice nel caso di collassamento a semplice degenero. In concreto si introduce una condizione (volumetrica, se si vuole chiamarla in qualche modo)

applicabile ai semplici:

$$\begin{cases} NV = \sqrt[n]{\frac{1}{n!} \det \begin{pmatrix} x_1 - x_0 & x_2 - x_0 & \dots & x_n - x_0 \end{pmatrix}} \\ \text{se } NV < \varepsilon \text{ viene terminato il ciclo di iterazioni} \end{cases} \quad (6.2)$$

dove $x_0 = x_l$ (ma poteva essere un vettore a scelta tra i vertici) e ε è un numero positivo legato alla precisione richiesta. Questa condizione esplica perfettamente il compito di terminare il ciclo di iterazioni. L'affiancamento di altre condizioni è risultato solo peggiorativo. L'unico difetto è il costo computazionale, poiché necessita il calcolo del determinante di una matrice $n \times n$, che viene risolto mediante la classe LU. Uno studio dell'analisi di complessità delle iterazioni dell'algoritmo 7 ha dimostrato come sia deleterio utilizzare una condizione di terminazione che abbia un costo computazionale elevato. Si è scelto quindi di attuare questa condizione di terminazione ogni tot iterazioni in modo da non calcolare troppe volte la condizione. Anche se il ciclo di iterazioni termina dieci iterazioni dopo rispetto al previsto, il risparmio computazionale nel caso di un numero notevole di iterazioni già avvenute è consistente. Il numero di iterazioni tra i calcoli della condizione dovrebbe essere scelto in base al costo computazionale della funzione da minimizzare. In realtà si è operato un tuning sulla classe di problemi scelti risultando approssimativamente conveniente operare la condizione ogni dieci iterazioni (vi sarà l'opportunità di un'implementazione più fine).

Modalità di ripetizione dell'algoritmo Per ogni ciclo di iterazioni, fino al termine dato dalla verifica della condizione, si ottiene un vettore a cui corrisponde l'attuale valore minimo della funzione. Per scongiurare il fatto che l'algoritmo sia terminato per collassamento del semplice non puntuale, l'algoritmo viene ripetuto con un semplice più piccolo (supponendo che si sia raggiunta la vicinanza con il punto di minimo) e una condizione di terminazione più stringente. Se l'algoritmo fa risultare un vettore abbastanza vicino a quello precedentemente calcolato:

$$\|x_{nuovo} - x_{precedente}\|_2 < \varepsilon$$

si registra il vettore migliore e si va al passo successivo. Se invece non si verifica questa condizione si riparte dal punto migliore dei due e si ricalcola un nuovo punto (il vettore conservato $x_{precedente}$ è naturalmente sempre sostituito con il migliore).

Nel passo successivo si genera un punto casuale (generazione di numeri pseudo-random estratti utilizzando l'orario corrente) e si ripete la procedura finora esposta ottenendo così una serie di vettori in numero uguale al numero di tentativi impostati. Non esiste un algoritmo che dia la certezza di ottenere il minimo globale di una funzione non convessa. Naturalmente la probabilità di ottenere il minimo globale aumenta con il numero di tentativi. Anche in questo caso mediante un'operazione di tuning si è scelto il numero di tentativi ottimale in relazione alla complessità del problema (sostanzialmente il numero di variabili).

6.2.4 Algoritmo Nelder-Mead modificato per la ricerca vincolata del minimo

Le modifiche applicate all'algoritmo si basano sulle teorie esposte nella sezione 4.5 a pagina 36. La strategia segue la teoria ma la applica in modo euristico, in similitudine con il funzionamento generale dell'algoritmo. La funzione minimizzata diventa quindi 4.21 a pagina 37:

$$\bar{L}(x, \lambda) = E(x) + \sum_{j=1}^m \lambda_j g_j^+(x) \quad (6.3)$$

Inizialmente il vettore λ (parametri di penalizzazione) è posto uguale al vettore nullo. Al termine di ogni trasformazione del semplice so si procede con la seguente procedura:

Dato il nuovo vertice x^{new} , derivato dalla trasformazione (quello che sostituisce x_h - anche nel caso di contrazione totale in cui vengono sostituiti più punti), il punto migliore presente x^{best} e λ^k riferito alla k -esima iterazione:

$$\text{se } \bar{L}(x^{new}, \lambda^k) \leq \bar{L}(x^{best}, \lambda^k),$$

$$\lambda_i^{k+1} = \lambda_i^k + s \max(0, g(x^{new})), \text{ per } i = 1, \dots, m,$$

$$x_{best} = \arg \left[\min_{x \in \{x^{new}, x^{best}, \text{altri vertici}\}} \bar{L}(x^{new}, \lambda^{k+1}) \right]$$

In questo modo al di fuori del rispetto delle condizioni di vincolo la funzione cresce progressivamente; L'aggiornamento dei parametri di penalizzazione porta a seguire euristicamente lo pseudo-gradiente del problema duale. Il parametro s deve essere scelto in modo appropriato, in relazione con la funzione $E(x)$. Operando un tuning è risultata la scelta del valore 0.5 per quest'ultimo parametro. Un parametro s troppo grande può risultare nocivo nel caso vada ad influenzare la condizione di termine (soprattutto sulle condizioni di terminazione basati sui valori della funzione). Fortunatamente la condizione di terminazione scelta non è basata sui valori della funzione ma sulla disposizione dei vertici del semplice, quindi presenta un buon grado di indipendenza dai valori della funzione. Ne viene quindi che è possibile scegliere un valore abbastanza grande del parametro s . Un valore grande permette di riportare all'interno delle condizioni vincolanti i vertici del semplice più velocemente (non c'è il rischio di ottenere risultati al di fuori delle condizioni imposte). Le funzioni di vincolo e la funzione da minimizzare sono state scalate in modo tale da avere ordini di grandezza identici (stesso peso dei vincoli e della funzione nel calcolo).

7 Schematizzazione del problema specifico e risultati numerici

Il problema reale di riferimento consiste come si è detto in una stazione di scambio termico che è costituita da due treni di scambio paralleli. La problematica

sottostà alle condizioni espresse in 3.1 a pagina 16 . Riprendendo lo schema 3.3 a pagina 23 si è introdotta una numerazione opportuna dei componenti presenti, evidenziando i punti di frazionamento (split). Si ricorda che gli split e gli scambiatori sarò denominati genericamente “strutture”. Gli split presenti sono 6, 5 esterni e 1 interno (connesso ad altre strutture). L’ottimizzazione ha come obiettivo:

- di massimizzare la temperatura di mescolamento adiabatico ottenuta dal mescolamento dei due flussi (linee 1) in uscita dagli scambiatori 6 e 15;
- in seconda istanza di ottenere determinate duties ai pumparounds, date dalla somma delle singole duties di ogni coppia a due a due (1 sommata a 8) , (3 sommata a 11) , (5 sommata a 14) sulle linee 0 (fluidi “caldi”).

Ciò equivale a massimizzare il recupero termico nella carica (la temperatura di ingresso della carica è un parametro indipendente). Nello schema si può osservare come per ogni struttura siano identificate due linee di passaggio, una identificata dall’indice 0, l’altra dall’indice 1. Si è scelto per motivi pratici di numerare negli scambiatori la linea del fluido che cede calore con 0 e la linea del fluido che assorbe il calore con 1. Negli split la numerazione è stata scelta identificando come treno 0 il treno inferiore nello schema e come treno 1 quello superiore; Il flusso uscente dallo split e diretto verso il treno 0 è identificato appunto con questo numero, analogamente per il flusso diretto al treno 1. Le portate sono volumetriche, calibrate su una densità di riferimento (a 15°C), moltiplicando per questa grandezza comunque si risale alla portata ponderale. Nello schema è possibile notare i misuratori:

- di temperatura, identificati da TI;
- di portata, identificati da FC;

I primi sono presenti in quasi tutte le sezioni di uscita e di ingresso degli scambiatori, tranne che nella linea 0 tra gli scambiatori 9 e 12, e nelle linee 0 e 1 tra gli scambiatori denominati entrambi 2; Quest’ultimi infatti sono in serie e adiacenti, quindi si è provveduto a considerarli come un singolo scambiatore evitando

la necessità delle misure intermedie; Risulta quindi assente l'unica misura di temperatura tra 9 e 12, linea 0; Le portate sono invece conosciute ovunque. Alcune delle misure sono ridondanti. Per la temperatura ciò risulta nella coppia di misuratori ("LL20TI253", "LL20TI203") le cui portate provengono da uno stesso split (16) e quindi dovrebbero indicare lo stesso valore di temperatura. Effettivamente presentano valori molto vicini, comunque si è provveduto a considerare la media dei due valori. Stesso discorso vale per la coppia di misuratori ("LL20TI235", "LL20TI287") le cui portate provengono dallo split 17 (sempre considerata la media). Per quanto riguarda i misuratori di portata si riscontra come il misuratore "20FC243" sia superfluo, poiché la portata misurata entra nello split 16 e già si conoscono le portate da "20FC263" e "20FC203" uscenti dallo split. La somma dei valori forniti da questi ultimi dovrebbe risultare identica a quella misurata da "20FC243", in realtà questa è leggermente inferiore (circa di un'unità). Si possono quindi attuare due strategie, o considerare le due portate in uscita (necessariamente utilizzate negli scambiatori in successione) e utilizzare la loro somma, oppure mediare le tre diverse misure mantenendo costante il rapporto tra le due portate in uscita. Nelle prove di calcolo effettuate in ambito di questa tesi si è scelto di utilizzare la prima strategia, ma è sempre possibile scegliere la seconda in fase di utilizzo del software. L'utilizzo delle misure di campo risulta fondamentale per il funzionamento del software.

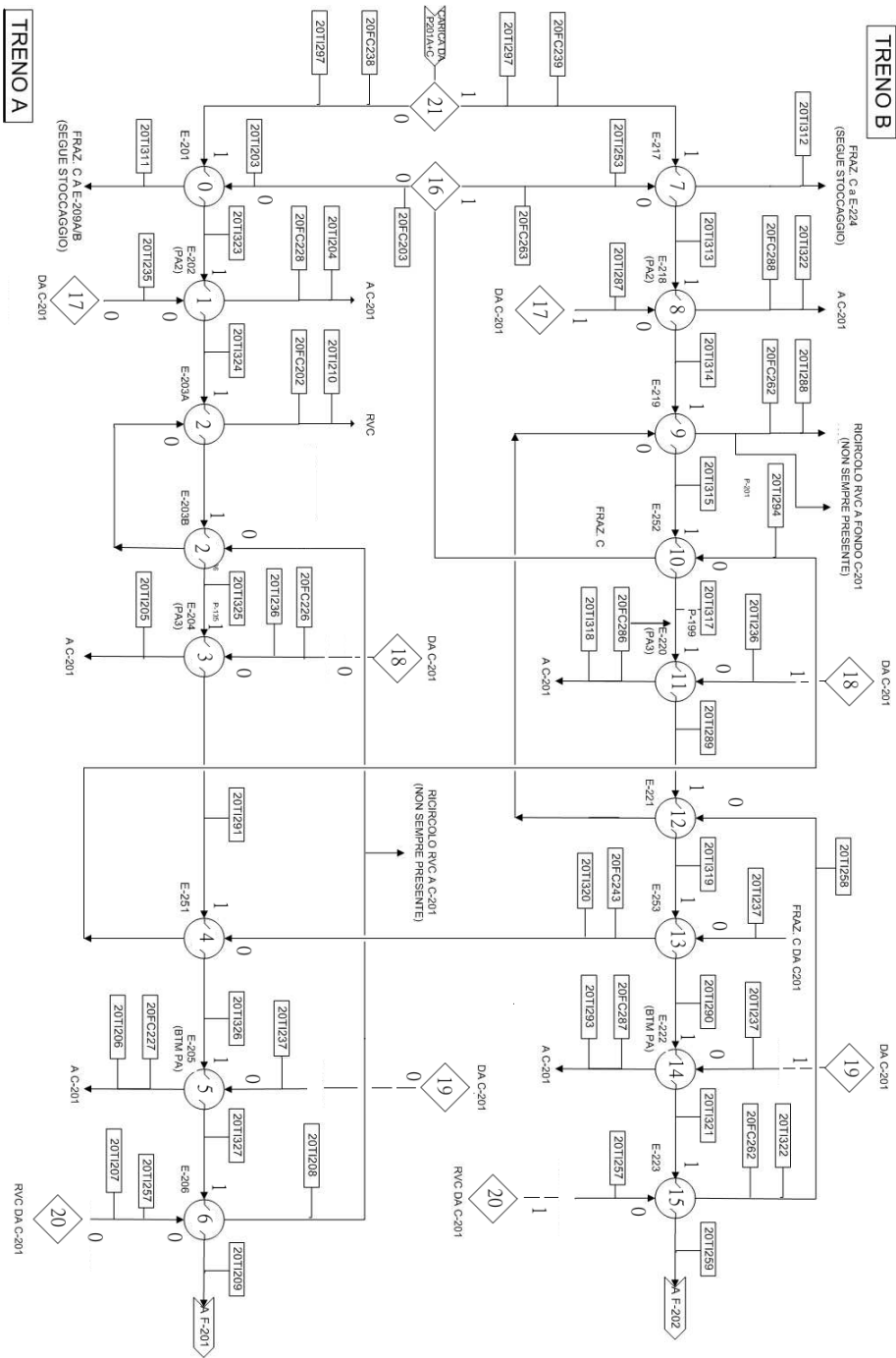


Figura 7.1: schema con numerazione

Il software verrà applicato sul problema anche nella modalità di imporre dei vincoli precisi sui flussi di calore. I vincoli imposti riguardano i flussi di calore totali scambiati dalle frazioni A, B e C (split 17, 18, 19). Queste duties imposte sono dovute al fatto di voler mantenere inalterata nell'ottimizzazione la qualità dei prodotti in uscita dalla colonna. I valori possono riferirsi o a quelli calcolati in un parallelo lavoro di tesi sulla dinamica della colonna, o nel mantenimento delle duties attuali al momento dell'ottimizzazione (esse vengono imposte dal controllore avanzato). Tra le esigenze espresse dal personale ENI vi è la necessità di non variare eccessivamente le portate tramite il sistema di controllo. È anche stata espressa la volontà di ricercare la condizione migliore in questo determinato range massimo di variazione, in questo modo le approssimazioni fatte sul modello risultano meno penalizzanti. L'algoritmo verrà eseguito, quando entrerà in funzione, ogni circa 3-4 ore (tempo necessario per esaurire i transitori), e migliorerà passo dopo passo il recupero termico, fino al raggiungimento del punto stazionario reale (che poi sarà comunque aggiornato nel tempo). Esso opererà nel tempo sopperendo al cambiamento delle condizioni (es. qualità della carica diversa) e sarà utilizzato anche per monitorare lo stato degli scambiatori (parametro AU - area per coefficiente globale di scambio).

7.1 Problematiche nell'utilizzo del software nell'applicazione sul problema di riferimento

Nell'applicazione del tool si riscontra una problematica dovuta alla presenza di by-pass a cavallo degli scambiatori. Questi by-pass si trovano in corrispondenza degli scambiatori collegati ai pumparounds, quindi le frazioni A, B e C, uno per ogni scambiatore. Con ciò la portata effettiva che circola negli scambiatori non è misurata, ma si conosce solo la portata prima del bivio by-pass scambiatore. Per risalire alla portata il personale ENI ha scelto di utilizzare delle considerazioni termodinamiche (conservazione dell'energia) e una ipotizzata proporzionalità nelle due linee tra la portata che passa nello scambiatore e la portata totale di linea. Da

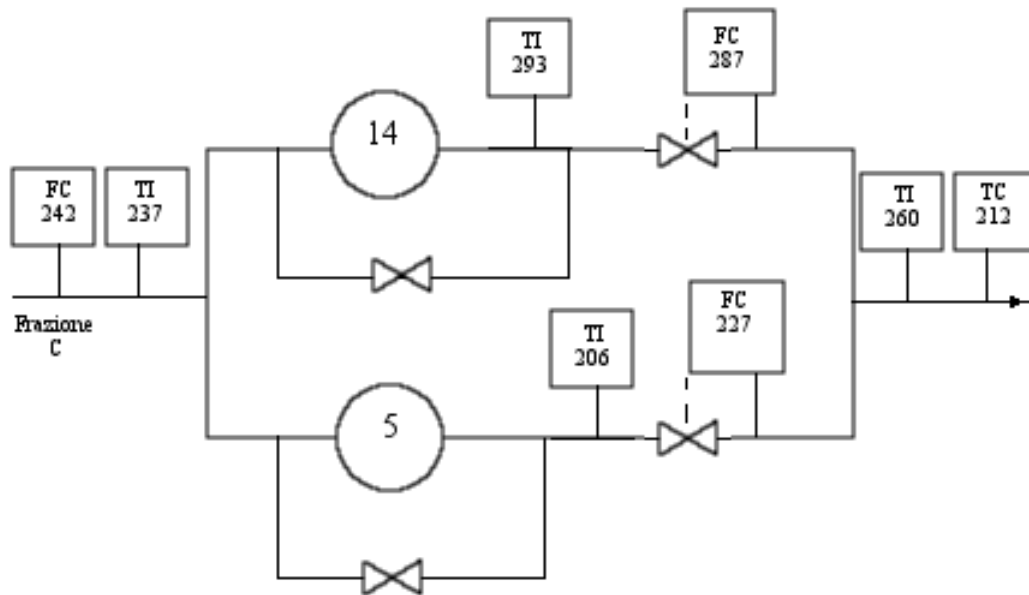


Figura 7.2: schema degli scambiatori della frazione C con i vari by-pass

ciò si può ottenere quindi una portata stimata. Questa problematica è sicuramente un aspetto delicato nell'applicazione. Infatti non solo va a incidere sul calcolo, ma anche sull'applicazione delle variazioni delle portate (si deve nuovamente ritornare alle portate a monte). La relazione per il calcolo della portata è data quindi da (si riportano solo le equazioni riferite alla frazione C - in collegamento con lo schema):

$$\begin{cases} (FC287 + FC227)(TC212 - TI237) = F_1(TI293 - TI237) + F_2(TI206 - TI237) \\ \frac{F_1}{FC287} = \frac{F_2}{FC227} \end{cases}$$

7.2 Risultati dall'applicazione del software su dati storici registrati

Il software è stato già applicato in tempo reale su un server secondario (ancora non interviene nel controllo avanzato), dando dei risultati plausibili e un risparmio compatibile con le attese (affermato dal personale ENI). In questa sezione si riportano dei dati a titolo di esempio e di valutazione di alcuni test sui dati storici registrati. I dati si riferiscono agli ultimi mesi di attività della colonna. In alcuni periodi di questo lasso di tempo sono presenti delle mancanze di dati a causa del non funzionamento (o mal funzionamento, manutenzione) di alcuni misuratori. Sono stati quindi selezionati delle registrazioni che avessero tutti i dati necessari. Sarà esposto un caso dettagliato dove si mostreranno tutte le quantità significative calcolate e i risultati relativi al risparmio energetico:

- ottimizzazione libera senza range di variazione delle variabili imposto.

In seguito verranno esposte solo le grandezze più significative del caso precedente con le seguenti modalità:

- ottimizzazione libera con range imposto (10% di scostamento delle variabili);
- ottimizzazione vincolata con range non imposto (duties fissate rispetto alle condizioni iniziali);
- ottimizzazione vincolata come in precedenza ma con range imposto (10%);

Dopo questa iniziale panoramica saranno riportati i dati relativi al recupero termico per diversi set di misurazioni (per diverse date e orari). Si avrà così una stima delle potenzialità e dell'ottenimento del risparmio.

7.2.1 Risultati relativi a un set di misure per la verifica del funzionamento generale

Il caso considerato è stato registrato in data 26/12/2012 (ore 6:00:00) e presenta i seguenti dati in input:

0 Tout[0]	140.5600	9 Tout[0]	208.5100	split 1 M[0]	88.8300
0 Tout[1]	114.1200	9 Tout[1]	163.3300	split 1 M[1]	49.6600
1 Tout[0]	177.7000	10 Tout[0] m	193.6500	split 1 Tin m	238.3700
1 Tout[1]	152.2700	10 Tout[0] m	192.0700	split 1 Tin m	238.0500
2 Tout[0]	211.9900	10 Tout[1]	188.1800	split 2 M[0]	64.4400
2 Tout[1]	182.1900	11 Tout[0]	225.3700	split 2 M[1]	61.2300
3 Tout[0]	205.2200	11 Tout[1]	220.2200	split 2 Tin	278.9400
3 Tout[1]	205.5600	12 Tout[1]	235.0200	split 3 M[0]	71.4900
4 Tout[0]	237.4800	13 Tout[0]	268.6000	split 3 M[1]	45.2200
4 Tout[1]	227.7800	13 Tout[1]	264.2300	split 3 Tin	324.3100
5 Tout[0]	276.4800	13 M[0]	69.5700	split 4 M[0]	40.1100
5 Tout[1]	255.4200	13 Tin[0]	324.3100	split 4 M[1]	46.1200
6 Tout[0]	279.4000	14 Tout[0]	278.7900	split 4 Tin	345.3300
6 Tout[1]	276.6700	14 Tout[1]	276.9900	split 5 M[0]	89.9200
7 Tout[0]	125.9400	15 Tout[0]	301.0500	split 5 M[1]	114.6900
7 Tout[1]	113.4300	15 Tout[1]	298.5500	split 5 Tin	91.9400
8 Tout[0]	144.8300	split 0 M [0]	31.0000		
8 Tout[1]	135.0500	split 0 M[1]	31.0000		

Le grandezze con accanto “m” e adiacenti sono state sintetizzate in un valor medio, come accennato in precedenza.

Ottimizzazione libera senza range In primo luogo si riportano i risultati ottenuti nel caso di ottimizzazione non vincolata, senza limiti di range. I dati fanno riferimento allo schema 7.1 . Si noti come le portate iniziali delle frazioni A, B e C

siano ridotte rispetto a quelle riportate dai misuratori nella tabella precedente. Ciò è dovuto dai bypass, e al calcolo tramite delle correlazioni delle portate effettive.

N SC	linea	m iniziali	m finali	DM %
0	0	31.000	23.182	-25.220
0	1	89.920	72.504	-19.368
1	0	50.379	37.917	-24.737
3	0	40.711	32.755	-19.542
5	0	49.539	38.288	-22.711
6	0	40.110	32.800	-18.226

N SC	linea	m iniziali	m finali	DM %
7	0	31.000	38.818	25.220
7	1	114.690	132.106	15.185
8	0	28.164	40.627	44.249
11	0	38.683	46.639	20.566
14	0	31.335	42.586	35.905
15	0	46.120	53.430	15.851
13	0	62.000	62.000	0.000

N SC	FRC iniziali	FRC finali
16	0.5000	0.3739
17	0.6414	0.4827
18	0.5128	0.4126
19	0.6125	0.4734
20	0.4652	0.3804
21	0.4395	0.3544

Si vede come per il raggiungimento del punto di massimo si hanno variazioni sostanziali delle portate, sbilanciandole sul treno con più scambiatori (come ci si poteva aspettare).

N SC	Ramo	Tout iniziali	Tout finali
0	0	140.56	130.24
0	1	114.12	116.08
1	0	177.70	167.71
1	1	152.27	157.57
2	0	211.99	207.49
2	1	182.19	190.36
3	0	205.22	205.84
3	1	205.56	213.48
4	0	237.48	240.24
4	1	227.78	234.89
5	0	276.48	272.78
5	1	255.42	263.44
6	0	279.40	280.34
6	1	276.67	284.68

N SC	Ramo	Tout iniziali	Tout finali
7	0	125.94	133.38
7	1	113.43	112.27
8	0	144.83	160.55
8	1	135.05	134.79
9	0	208.51	214.95
9	1	163.33	160.90
10	0	192.86	191.62
10	1	188.18	184.41
11	0	225.37	228.79
11	1	220.22	215.81
12	0	266.96	268.61
12	1	235.02	229.96
13	0	268.60	264.41
13	1	264.23	257.22
14	0	278.79	280.74
14	1	276.99	271.63
15	0	301.05	301.01
15	1	298.55	293.33

Le temperature sono influenzate moderatamente dalla variazione di portate. Si nota anche che vi è una diminuzione di temperatura nel treno di maggiori dimensioni e un aumento nell'altro (risultato abbastanza ragionevole).

Ent_ott [Kcal / h]	Ent_in K [Kcal / h]	Guadagno [Kcal / h]	Guadagno [KW]	Tfin_Iniziale	Tfin_ottim
38413746.0908	38188461.1871	225284.9037	261.9563	289.0457	290.2842

L'incremento di temperatura di mescolamento adiabatico si attesta a poco più di un grado centigrado. L'incremento del recupero termico risulta secondo il personale ENI in linea con le aspettative considerando la modalità dell'intervento. Nonostante aumenti solo di un grado la temperatura si può stimare un risparmio

annuo di circa 100 tonnellate di gas naturale. Ciò si spiega con il fatto che le portate in gioco sono notevoli.

Ottimizzazione libera con range Per quanto riguarda le portate si nota come effettivamente siano stati rispettati i range imposti (10%)

N SC	linea	m iniziali	m finali	DM %
0	0	31.000	27.901	-9.998
0	1	89.920	80.928	-10.000
1	0	50.379	47.563	-5.590
3	0	40.711	36.843	-9.502
5	0	49.539	46.405	-6.325
6	0	40.110	36.099	-10.000

N SC	linea	m iniziali	m finali	DM %
7	0	31.000	34.099	9.998
7	1	114.690	123.682	7.840
8	0	28.164	30.981	10.000
11	0	38.683	42.551	10.000
14	0	31.335	34.469	10.000
15	0	46.120	50.131	8.697
13	0	62.000	62.000	0.000

Il recupero termico risulta leggermente più contenuto, ma non eccessivamente rispetto al caso precedente:

Ent_ott [Kcal / h]	Ent_in K [Kcal / h]	Guadagno [Kcal / h]	Guadagno [KW]	Tfin_Iniziale	Tfin_ottim
38348171.2088	38188461.1871	159710.0217	185.7073	289.0457	289.9240

Si vede come si è recuperato un grado anche in questo caso. Ciò fa pensare che la funzione sia abbastanza piatta vicino al punto di massimo.

Ottimizzazione vincolata senza range imposto L'ottimizzazione vincolata viene operata secondo quanto stabilito in accordo dal personale ENI. È stato scelto di mantenere vincolate le duties rilevate nel sistema iniziale prima dell'ottimizzazione. Questo dovrebbe permettere di mantenere inalterata la qualità dei prodotti. La tolleranza sul vettore delle duties è stata posta pari al 1% (equazione 4.18).

N SC	linea	m iniziali	m finali	DM %
0	0	31.000	22.684	-26.827
0	1	89.920	74.783	-16.834
1	0	50.379	26.824	-46.756
3	0	40.711	39.512	-2.945
5	0	49.539	38.870	-21.535
6	0	40.110	31.820	-20.669

N SC	linea	m iniziali	m finali	DM %
7	0	31.000	39.316	26.827
7	1	114.690	129.827	13.198
8	0	28.164	51.720	83.635
11	0	38.683	39.882	3.099
14	0	31.335	42.004	34.046
15	0	46.120	54.410	17.975
13	0	62.000	62.000	0.000

È interessante notare come le portate siano variate in modo sostanzialmente differente rispetto al caso di ottimizzazione libera. Anche in questo caso il recupero termico risulta più contenuto, ma sostanzialmente vicino al valore di ottimizzazione libera:

Ent_ott [Kcal / h]	Ent_in K [Kcal / h]	Guadagno [Kcal / h]	Guadagno [KW]	Tfin_Iniziale	Tfin_ottim
38363467.2362	38188461.1871	175006.0491	203.4931	289.0457	290.0080

L'errore sul vettore duties risulta pari all'1%, quindi la soluzione si trova sul confine dei vincoli. Si è osservato anche come le duties si discostino abbastanza in maniera omogenea e quindi il tipo di vincolo imposto sia efficace.

Ottimizzazione vincolata con range imposto Questa è la modalità vera e propria per applicare la libreria al sistema reale. Anche in questo caso la tolleranza sul vettore delle duties è stata posta pari al 1% (equazione 4.18).

N SC	linea	m iniziali	m finali	DM %
0	0	31.000	27.900	-10.000
0	1	89.920	81.455	-9.414
1	0	88.830	84.615	-4.745
3	0	64.440	64.638	0.307
5	0	71.490	66.968	-6.325
6	0	40.110	36.099	-10.000

N SC	linea	m iniziali	m finali	DM %
7	0	31.000	34.100	10.000
7	1	114.690	123.155	7.381
8	0	49.660	53.875	8.488
11	0	61.230	61.032	-0.324
14	0	45.220	49.742	10.000
15	0	46.120	50.131	8.697
13	0	62.000	62.000	0.000

I range vengono rispettati e si ottiene sempre un risparmio consistente, il minore tra tutte le prove.

Ent_ott [Kcal / h]	Ent_in K [Kcal / h]	Guadagno [Kcal / h]	Guadagno [KW]	Tfin_Iniziale	Tfin_ottim
38314033.0774	38188461.1871	125571.8903	146.0122	289.0457	289.7364

Le duties prima e dopo l'ottimizzazione risultano:

	Duties Totali Iniz [Kcal/h]	Duties Totali Ott [Kcal/h]
fraz A	3192773.558	3238657.608
fraz B	3087698.208	3067914.891
fraz C	2515759.382	2551894.499

Si nota lo scostamento maggiore (1.5%) nella frazione B rispetto alle altre, risulta comunque un valore contenuto in confronto all'errore generale imposto (1%).

Si sottolinea infine che ripetendo più volte il calcolo il risultato ottenuto è sempre vicinissimo al precedente e considerando l'aleatorietà introdotta nell'algoritmo, si può affermare che il risultato converge molto probabilmente nel minimo assoluto ricercato.

7.2.2 Verifica del recupero termico su più set di dati

Si è quindi proceduto a valutare, in diverse condizioni storizzate nei dati delle misurazioni, come può evolvere il recupero termico anche in relazione a funzionamenti non ottimali in determinati periodi di attività. Si è scelto di prelevare i set di dati in un periodo che va dal 03/31/2013 al 17/05/2013 (funzionamento ottimale dei misuratori).

I risultati dell'applicazione del software sono sintetizzati nei grafici sottostanti. Si può osservare come il guadagno e la differenza di temperatura di mescolamento siano abbastanza costanti nel tempo. Questo è un dato confortante poiché sembra confermare la non eccessiva sensibilità del software rispetto ai disturbi dei misuratori. Si vede anche come il guadagno segua in modo consistente l'andamento

della temperatura adiabatica calcolata prima dell'ottimizzazione. Va sottolineato che l'ottimizzazione viene effettuata variando le portate in un range piccolo (10%) quindi è possibile che dopo il primo aggiustamento si possa raggiungere condizioni ancora più favorevoli.

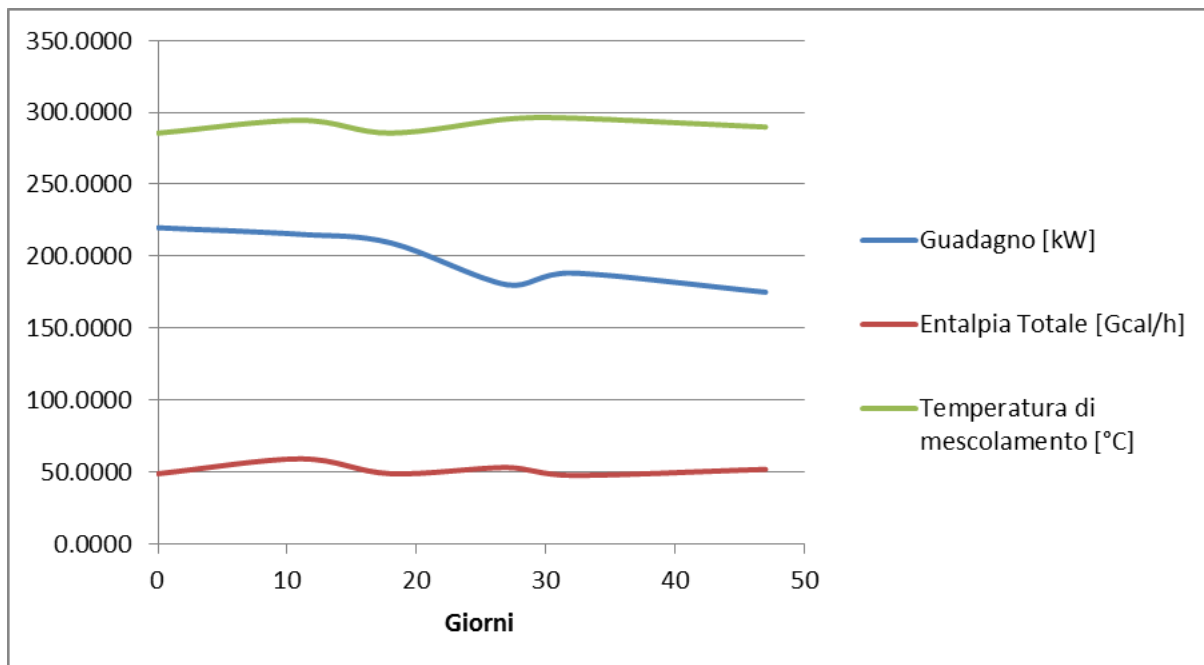


Figura 7.3: andamento delle variabili caratteristiche

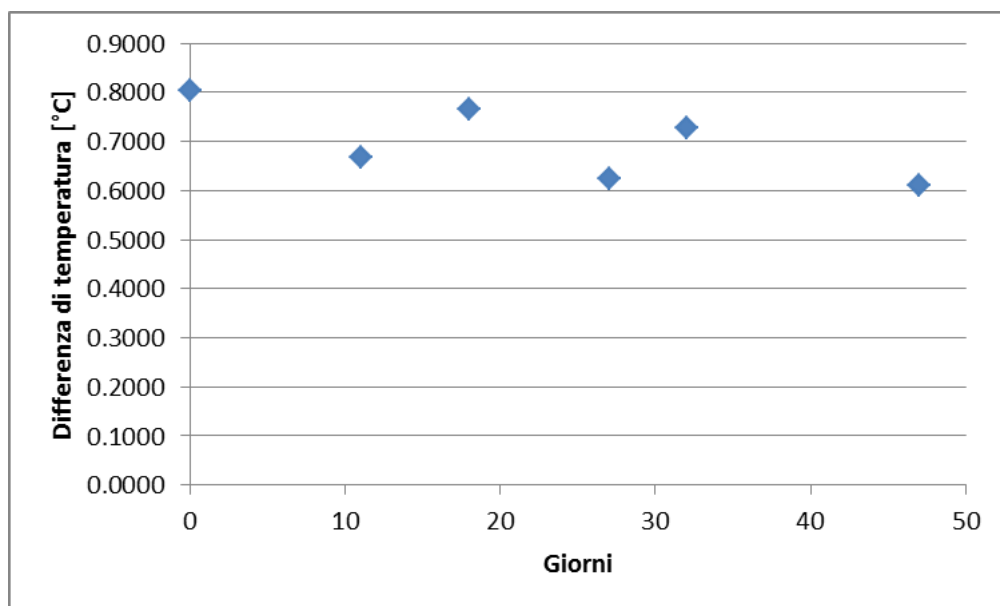


Figura 7.4: andamento della differenza di temperatura di mescolamento tra prima e dopo l'ottimizzazione

Oltre a questi test se ne sono attuati altri in sede della Raffineria di Livorno, su un server inattivo ma che presenta la possibilità di accedere alle misure di campo in tempo reale. I risultati hanno mostrato come la tendenza qui sopra esposta sia effettivamente riscontrata. Le prove hanno dimostrato la robustezza del software e la ragionevolezza dei risultati ottenuti.

8 Prospetto informativo per l'utilizzo della libreria OpTExLib

Il software consiste in una libreria eseguibile ATL (*Active Template Library*) COM (Component Object Model) che permetta la creazione di un set di classi C++ da parte di programmi in VBA (Visual Basic) o VC++ (Visual C++). Questo permette di poter usufruire della versatilità (in fase di realizzazione del codice) e della velocità garantite dallo standard C++. Le funzioni che consentono l'interfaccia

tra la libreria e il codice che la esegue (funzioni della classe server) si possono dividere in diverse categorie:

1. **funzioni per l'impostazione generale** del sistema da analizzare (es. creazione e reset del sistema, esecuzione del calcolo, numero di scambiatori, numero di split, precisione cercata, limiti nelle variabili di ricerca ecc.);
2. **funzioni di input** della struttura del sistema e delle proprietà dei flussi presenti (temperature, portate, caratteristiche fisiche dei flussi che coinvolgono negli scambiatori e negli split);
3. **funzioni di output** per l'ottenimento dei risultati o di altre caratteristiche volute (portate calcolate, verifica del rispetto delle condizioni imposte ecc.);

In seguito verranno riportate tutte le funzioni del server della libreria con la relativa descrizione. A queste funzioni sono associate altre funzioni, membri della classe principale ottimizzatrice di cui viene istanziato un oggetto al momento del richiamo della libreria (classe optimizer, file optimizer.h - a cui appartengono tutte le funzioni che permettono l'interfaccia). Successivamente verrà fornito l'ordine opportuno per il richiamo delle funzioni per l'impostazione, esecuzione e accesso ai risultati per una ipotetica una stazione di scambio termico. Ci si riferisce a *duties* come flusso di calore scambiato.

Per quanto riguarda la schematizzazione e gli input si deve tenere conto che:

- le portate devono essere inserite in [m^2/h], le temperature in [$^{\circ}\text{C}$], le duties in [kcal/h], i calori specifici in [$\text{kcal}/(^{\circ}\text{C kg})$] e le densità in [kg/m^3];
- si denominano strutture i componenti della stazione di scambio termico (scambiatori o split), nel caso in cui si voglia specificare si nomineranno questi ultimi;
- ogni struttura è identificata da un numero intero, la numerazione parte da (0) per il primo scambiatore fino a (numero totale di scambiatori-1), gli split sono convenzionalmente numerati dal numero (numero totale di scambiatori)

fino a (numero di scambiatori + numero di split - 1), si tratta comunque di un sistema di identificazione, non è necessario avere un determinato ordine o una numerazione collegata alla disposizione degli scambiatori nello schema (è solo importante ricordare da quando parte la numerazione degli split) - ESEMPIO se si hanno 4 scambiatori e 3 split gli scambiatori avranno i numeri [0,1,2,3] e gli split [4,5,6] in totale risultano quindi 7 strutture e la numerazione di queste termina a 6;

- uno scambiatore di calore (senza miscelazione) presenta due linee (*line* negli argomenti delle funzioni) di passaggio, uno in cui scorre il fluido freddo e una in cui scorre il fluido caldo;
- non è necessario specificare quale sia il fluido freddo e quale il fluido caldo, è importante tenere presente che due linee sono identificate da un intero che assume il valore 0 o 1;
- gli split presentano una linea di ingresso identificata per convenzione dal valore 0 nell'argomento *line* e due linee di uscita, di queste ultime l'inserimento degli input si ottiene esattamente come negli scambiatori, con le linee identificate da un intero che assume valore 0 o 1;
- le connessioni da fornire sono solo in ingresso a ogni struttura;
- attualmente l'impostazione delle duties è concepita per due treni di scambio, la duty viene impostata sugli split di ingresso voluti (quelli che non hanno connessioni in ingresso) e il programma provvede ad ottenere le stesse come somma totale dei flussi di calore scambiati in tutti gli scambiatori percorsi dalle linee uscenti dallo split (questa parte potrà essere sottoposta a miglioramenti successivi);

In alcuni degli argomenti delle funzioni sono segnalati i possibili valori da inserire tra parentesi quadre (es. [0,1]).

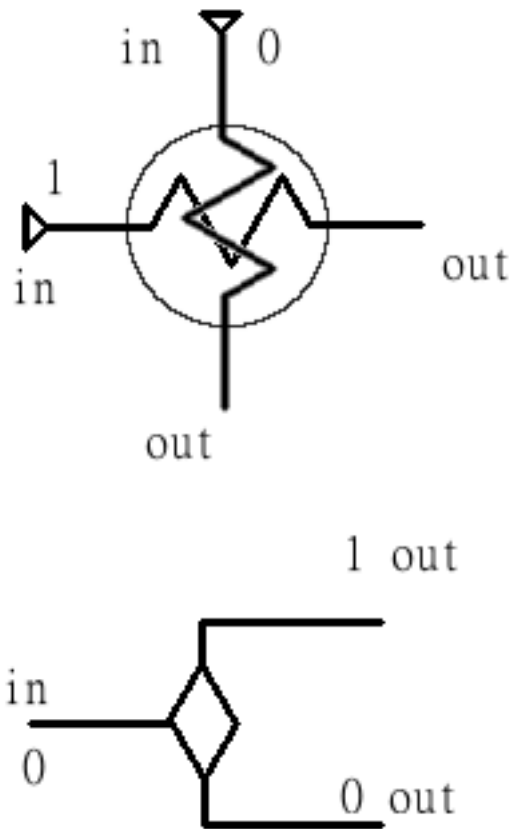


Figura 8.1: strutture (scambiatore e split) componenti il sistema di scambio termico virtuale.

8.1 Funzioni per l'impostazione generale

8.1.1 Create_Class

Funzione: Create_Class(LONG n_scamb, LONG n_split)

argomenti:

- n_scamb : numero di scambiatori
- n_split : numero di split

output: LONG 1 , restituisce questo valore se la funzione è stata eseguita correttamente;

descrizione: imposta i parametri della struttura creando lo spazio di memoria necessario all'acquisizione dei dati.

8.1.2 Reset

Funzione: Reset()

nessun argomento

output: LONG 1 , restituisce questo valore se la funzione è stata eseguita correttamente;

descrizione: cancella la memoria allocata ritornando alla situazione precedente alla creazione, utilizzabile se si vuole usufruire più volte della classe ottimizzatrice senza richiamare la libreria.

8.1.3 Set_Parameters

Funzione: Set_Parameters(DOUBLE prec, LONG maxfun, LONG n_att,DOUBLE err);

argomenti:

- prec : precisione desiderata (norma 2 sul vettore portate calcolato nell'algoritmo di ricerca del minimo);
- maxfun: numero massimo di valutazioni per ogni chiama dell'algoritmo di ricerca del minimo;
- n_att: numero di tentativi casuali in ordine della ricerca globale del minimo attuati ad ogni chiamata dell'algoritmo di ricerca;
- err: errore accettato sulle duties (norma 2 sul vettore duties - ovvero il vettore contenente le duties richieste in ottimizzazione vincolata);

output: LONG 1 , restituisce questo valore se la funzione è stata eseguita correttamente;

descrizione: imposta i parametri relativi all'algoritmo e alla precisione di ricerca sia sulle duties richieste che sulle portate finali calcolate.

8.1.4 Set_Obj

Funzione: Set_Obj(LONG S1, LONG line1, LONG S2 ,LONG line2);

argomenti:

- S1 : scambiatore su cui massimizzare la temperatura di uno dei flussi di uscita (presumibilmente appartenete al primo treno di scambio);
- line1: il flusso in uscita della linea scelta appartenente allo scambiatore S1 [0,1];
- S2: come S1 ma riferito al secondo scambiatore (secondo treno di scambio);
- line2: nello stesso modo di line1 ma riferito allo scambiatore S2 [0,1];

output: LONG 1 , restituisce questo valore se la funzione è stata eseguita correttamente;

descrizione: permette di impostare la funzione obiettivo, in questo caso si considerano due treni di scambio, quindi permetterà di impostare come funzione obiettivo l'entalpia totale dovuta al mescolamento dei due fluidi scelti uscenti dai due scambiatori scelti (unica funzione di cui si necessita la modifica in caso di più di due treni di scambio).

8.1.5 Set_Range

Funzione: Set_Range(DOUBLE range);

argomenti:

- range : range delle variabili su cui poter attuare la ricerca;

output: LONG 1 , restituisce questo valore se la funzione è stata eseguita correttamente.

descrizione: permette che nel risultato finale ogni portata non abbia una variazione percentuale rispetto al dato iniziale maggiore del valore range;

8.1.6 Initialize

Funzione: Initialize();

nessun argomento

output: LONG 1, 2 o 3 :

- valore 1: il calcolo è andato a buon fine con il rispetto dei vincoli richiesti;
- valore 2: non sono stati forniti un numero sufficiente di dati per operare il calcolo;
- valore 3: il calcolo fornisce risultati ma non vi è stato modo di rispettare i vincoli richiesti nel campo di ricerca impostato;

descrizione: permette di inizializzare il calcolo e di verificare l'esito dello stesso.

8.2 funzioni di input

8.2.1 Set_Flows

Funzione: Set_Flows(LONG identif, DOUBLE flow_0, DOUBLE flow_1);

argomenti:

- identif : numero identificativo della struttura considerata
- flow_0 : valore della portata della prima linea;
- flow_1 : valore della portata della seconda linea;

output: LONG 1 , restituisce questo valore se la funzione è stata eseguita correttamente

descrizione: imposta i valori delle portate per una determinata struttura, se essa è equivalente a quella in uscita da un altro scambiatore (non è un dato di input) può essere impostata pari a -1; se si tratta di uno split inserire la portata in ingresso come argomento “flow_0” e (-1) in “flow_1”;

8.2.2 Set_Fluid

Funzione: Set_Fluid(LONG identif, LONG line, DOUBLE C_coef1, DOUBLE C_coef2, DOUBLE D_coef1, DOUBLE D_coef2);

argomenti:

- identif : numero identificativo della struttura;
- line: linea di cui si vogliono specificare le proprietà [0,1];
- C_coef1: termine noto della relazione lineare tra calore specifico a pressione costante e temperatura;
- C_coef2: coefficiente angolare della relazione lineare tra calore specifico a pressione costante e temperatura, unità di misura [kcal/(°C² kg)];
- D_coef1: termine noto della relazione lineare tra densità a pressione costante e temperatura;
- Dcoef_1: coefficiente angolare della relazione lineare tra densità a pressione costante e temperatura, unità di misura in [kg/(°C m³)]; ;

output: LONG 1 , restituisce questo valore se la funzione è stata eseguita correttamente

descrizione: imposta le relazioni lineari delle proprietà calore specifico e densità in funzione della temperatura, se non si desidera fornire i valori dei coefficienti angolari porli uguali a 0;

8.2.3 Set_Connections

Funzione: Set_Connections(LONG identif, LONG C0_0, LONG C0_1, LONG C1_0, LONG C1_1);

argomenti:

- identif : numero identificativo dello scambiatore o split considerato;
- C0_0 : numero identificativo struttura connessa alla linea 0;
- C0_1 : numero identificativo della linea della struttura in connessione C0_0;
- C1_0 : numero identificativo struttura connessa alla linea 1;
- C1_1 : numero identificativo della linea della struttura in connessione C1_0;

output: LONG 1 , restituisce questo valore se la funzione è stata eseguita correttamente;

descrizione: imposta i collegamenti in ingresso alla struttura “identif”, alla linea 0 è connessa la linea C0_1 della struttura C0_0, alla linea 1 è connessa la linea C1_1 della struttura C1_0. Se non vi è una delle due connessioni è necessario impostare ai valori della linea e del numero identificatore della connessione il valore -1.

8.2.4 Set_Tin

Funzione: Set_Tin(LONG idt, LONG line, DOUBLE temp, LONG flag_);

argomenti:

- identif : numero identificativo della struttura considerata;
- line : linea considerata;
- temp : valore della temperatura in ingresso;
- flag_ : validità della misura temperatura [0,1];

output: LONG 1 , restituisce questo valore se la funzione è stata eseguita correttamente;

descrizione: imposta i valori della temperatura di un flusso in ingresso nella linea di una determinata struttura, è presente anche un indice che comunichi la validità della misura fornita (nel caso di 0 non verrà impostata).

8.2.5 Set_Tout

Funzione: Set_Tout(LONG idt, LONG line, DOUBLE temp, LONG flag_);

argomenti:

- identif : numero identificativo della struttura considerata;
- line : linea considerata;
- temp : valore della temperatura in uscita;
- flag_ : validità della misura temperatura [0,1];

output: LONG 1 , restituisce questo valore se la funzione è stata eseguita correttamente;

descrizione: imposta i valori della temperatura di un flusso in uscita dalla linea di una determinata struttura, è presente anche un indice che comunichi la validità della misura fornita (nel caso di 0 non verrà impostata).

8.2.6 Set_Duty

Funzione: Set_Duty(LONG spl, DOUBLE D, LONG* flag)

argomenti:

- spl : numero identificativo dello split considerato
- D : valore della duty imposto allo split;

output: LONG, 1 , restituisce questo valore se la funzione è stata eseguita correttamente;

descrizione: permette di impostare la duty su un determinato split, ovvero il programma provvederà con l'esecuzione a garantire il rispetto del vincolo; la duty si riferisce alla somma totale dei flussi di calore scambiati dalla portata totale entrante nello split. La duty sarà da impostare positiva se il fluido della linea dovrà ottenere calore, negativa se dovrà cederlo.

8.3 Funzioni di output

8.3.1 Call_Flow

Funzione: Call_Flow(LONG idt, LONG line);

argomenti:

- idt : numero identificativo della struttura considerata;
- line : linea considerata;

output: DOUBLE, portata;

descrizione: restituisce il valore della portata per la data struttura e linea.

8.3.2 Call_Tout

Funzione: Call_Tout(LONG idt, LONG ramo);

argomenti:

- idt : numero identificativo della struttura considerata;
- line : linea considerata;

output: DOUBLE, temperatura in uscita;

descrizione: restituisce la temperatura in uscita per la data struttura e linea.

8.3.3 (OPZIONALE) Call_Tout_initial

Funzione: Call_Tout_initial(LONG idt, LONG ramo,DOUBLE* ris);

argomenti:

- idt : numero identificativo della struttura considerata;
- line : linea considerata;

output: DOUBLE, temperatura in uscita iniziale;

descrizione: restituisce la temperatura in uscita del sistema iniziale, ovvero prima che fosse attuata l'ottimizzazione (funzione definita per comodità di uso nell'interfaccia con Excel);

8.3.4 Call_Tin

Funzione: Call_Tin(LONG idt, LONG ramo);

argomenti:

- idt : numero identificativo della struttura considerata;
- line : linea considerata;

output: DOUBLE, temperatura d'ingresso;

descrizione: restituisce il valore temperatura in ingresso per la data struttura e linea.

8.3.5 Final_Enthalpy

Funzione: Final_Enthalpy();

nessun argomento;

output: DOUBLE, entalpia totale dopo l'ottimizzazione (mescolamento adiabatico);

descrizione: restituisce il valore dell'entalpia totale, ad ottimizzazione avvenuta, considerando la miscelazione adiabatica dei due flussi scelti mediante Set_Obj (è la funzione obiettivo di cui viene trovato il punto di massimo).

8.3.6 Initial_Enthalpy

Funzione: Initial_Enthalpy();

nessun argomento;

output: DOUBLE, entalpia totale prima dell'ottimizzazione (mescolamento adiabatico);

descrizione: restituisce il valore dell'entalpia totale, prima dell'ottimizzazione (sistema iniziale), considerando la miscelazione adiabatica dei due flussi scelti mediante Set_Obj (è la funzione obiettivo di cui viene trovato il punto di massimo).

8.3.7 final_Tout

Funzione: final_Tout();

nessun argomento;

output: DOUBLE, temperatura in uscita;

descrizione: restituisce il valore della temperatura in uscita, ad ottimizzazione avvenuta, del flusso ottenuto dalla miscelazione adiabatica dei due flussi scelti mediante Set_Obj (è la funzione obiettivo di cui viene trovato il punto di massimo).

8.3.8 initial_Tout

Funzione: final_Tout();

nessun argomento;

output: DOUBLE, temperatura in uscita;

descrizione: restituisce il valore della temperatura in uscita, prima dell'ottimizzazione, del flusso ottenuto dalla miscelazione adiabatica dei due flussi scelti mediante Set_Obj (è la funzione obiettivo di cui viene trovato il punto di massimo).

8.3.9 Gain

Funzione: Gain();

nessun argomento;

output: DOUBLE, differenza tra entalpia totale dopo l'ottimizzazione ed entalpia totale prima dell'ottimizzazione;

descrizione: restituisce il valore dell'incremento di entalpia della portata obiettivo scelta grazie all'ottimizzazione.

8.3.10 Error_Duty

Funzione: Error_Duty();

nessun argomento;

output: DOUBLE, $\text{norma2}(\text{vettore duties ricercato} - \text{vettore duties ottenuto}) / \text{norma2}(\text{vettore duties ricercato})$;

descrizione: restituisce la norma vettoriale (2) della differenza tra il vettore di duties cercato e il vettore duties effettivamente ottenuto divisa per la norma del vettore duties cercato.

8.4 Ordine di utilizzo delle funzioni, esempio teorico

Le funzioni sopra elencate devono essere utilizzate in un certo ordine. Supponendo ad esempio di avere una serie di scambiatori e di split si consiglia di attuare in questo modo l'impostazione, l'esecuzione e l'accesso ai dati:

1. creazione del sistema mediante "Create_Class" con cui si impostano le quantità di scambiatori e split;
2. impostazione della funzione obiettivo "Set_Obj" riferita ai flussi scelti di due scambiatori determinati;
3. eventuale impostazione del range di ricerca "Range";

4. eventuale impostazione dei parametri riguardanti il calcolo e l'algoritmo "Set_Parameters" (se se ne vuole modificare solo alcuni andranno comunque introdotti tutti);
5. completamento delle informazioni di ogni struttura facente parte del sistema presumibilmente in questo ordine:
 - (a) impostare le connessioni in ingresso (una se si tratta di split) se presenti;
 - (b) impostare le portate in ingresso nelle linee non connesse (strettamente necessario);
 - (c) impostare le temperature di ingresso ove non presenti connessioni in ingresso alle linee (strettamente necessario);
 - (d) impostare le temperature di uscita (solo in caso di struttura scambiatore);
 - (e) impostare le caratteristiche dei fluidi (calore specifico, densità) nelle linee non connesse;
6. opzionalmente impostare determinate duties su alcuni degli split presenti;
7. eseguire il calcolo mediante "Initialize";
8. ottenere i risultati con le funzioni di output nell'ordine che si preferisce;
9. opzionalmente resettare la classe mediante "Reset" ed effettuare nuovamente la procedura dal punto 1 se vi è la necessità di effettuare nuovamente un'altra esecuzione.

9 Conclusioni

In conclusione si può affermare che la realizzazione del software è pienamente riuscita. Il sistema presenta una certa versatilità, e può operare su stazioni di scambio termico con diversi schemi di connessione tra gli scambiatori. Il modello fisico-matematico è stato verificato nella sua validità, anche mediante un'analisi parametrica per valutazione delle approssimazioni imposte. Alle soluzioni termodinamiche ottenute si sono affiancate quelle calcolate da un simulatore di processo (Aspen HYSYS (Aspentech)) ed è stata verificata una forte congruenza dei risultati. Si è quindi operata l'applicazione su un sistema di riferimento, la stazione di scambio termico associata alla colonna Vacuum. Il sistema è stato testato innumerevoli volte, con diversi set di misure registrate relative a diversi periodi dell'anno in corso. I disturbi sui misuratori sembrano non pesare consistentemente sui risultati del software. Dall'applicazione della libreria si è riusciti ad avere una visione più profonda del problema. Sostanzialmente l'ottimizzazione può essere condotta con risultati tangibili, ma va sottolineato che le stazioni di scambio termico in cui si possono variare le ripartizioni, ma non i valori assoluti delle portate, risultano sistemi abbastanza rigidi. La funzione obiettivo del sistema di riferimento, data dall'entalpia totale dopo il mescolamento dei due flussi di carica in uscita dai due treni, presenta una forma piana nei pressi del punto di massimo e non permette grandi variazioni rispetto al valore iniziale calcolato. Il recupero termico è risultato comunque in linea con le aspettative del personale ENI. Il software verrà applicato in tempi brevi al controllo avanzato funzionante in tempo reale con esecuzioni cicliche. Provvederà quindi a calcolare le nuove ripartizioni di portate e a raggiungere l'ottimizzazione reale effettiva partendo dalla nuova condizione imposta volta per volta. Quindi esplicherà anche la funzione di:

- non alterare con l'ottimizzazione dei prodotti grazie all'imposizione e mantenimento di determinati flussi di scambio termico per alcune frazioni di interesse, denominati duties e ottenuti tramite lavori di analisi paralleli o dalle condizioni operative attuali (che presentano la qualità cercata);

- strumento di monitoraggio delle condizioni degli scambiatori (sporramento, necessità di manutenzione) in tempo reale;

Si è potuto osservare come le duties, quando imposte, vengono effettivamente mantenute in modo efficace. L'imposizione delle duties identiche a quelle del sistema in condizioni iniziali portano ad un recupero termico paragonabile all'ottimizzazione non vincolata. In concreto il sistema porterà con la prima esecuzione all'incremento di circa un grado di temperatura nella carica complessiva. Il risparmio equivale a circa cento tonnellate di gas naturale risparmiate ai forni. Considerando l'assenza di costi significativi nell'applicazione di questo strumento, si può affermare che il miglioramento previsto nelle prestazioni sia significativo.

Dopo l'effettiva applicazione nel sistema di riferimento (Vacuum), il sistema sarà applicato come è stato già descritto su altri impianti (colonna Topping e colonna di de-asfaltazione). La colonna Topping presenta delle portate in gioco maggiori. Ci si aspetta quindi in raggiungimento di risultati migliori in termini di risparmio (che dipende in modo significativo dalle portate). Non si presentano grandi complicazioni in questi diversi sistemi, poiché non è necessario ricompilare la libreria realizzata.

Per quanto riguarda la parte informatica e gli algoritmi implementati in questo lavoro si può affermare che il sistema presenta una robustezza notevole. Ogni misura deve essere introdotta solamente una volta e il sistema provvede a utilizzare le informazioni per creare un modello virtuale su cui operare il calcolo. Le soluzioni convergono con buona probabilità al punto di massimo globale ricercato.

Si deve infine sottolineare come il sistema possa operare su sistemi in cui si abbiano delle misurazioni precise della temperatura. Infatti gli incrementi di temperatura raggiunti si attestano su valori contenuti, quindi se i misuratori dovessero avere incertezze maggiori di questi valori il risultato sarebbe non attendibile e anche se veritiero sarebbe difficile verificarne l'attuazione. Il personale ENI assicura comunque che le misure di temperatura dovrebbero essere sufficientemente attendibili.

Si ricorda che questo documento presenta un prospetto informativo nella sezione 8. In esso vi sono tutte le istruzioni necessarie per un'applicazione corretta del software ed è stato redatto in accordo con le richieste espresse dal personale ENI.

A Analisi di sensibilità parametrica

A.1 Analisi degli effetti dovuti alla variazione del parametro f conseguentemente al cambiamento delle condizioni operative

Il parametro di correzione f (denominato poi FC) è definito come:

$$P = \frac{T_{c2} - T_{c1}}{T_{h1} - T_{c1}} \quad R = \frac{T_{h1} - T_{h2}}{T_{c2} - T_{c1}}$$

$$X = \frac{1 - \left(\frac{R P - 1}{P - 1}\right)^{\frac{1}{N}}}{R - \left(\frac{R P - 1}{P - 1}\right)^{\frac{1}{N}}}$$

$$F = \frac{\left(\frac{\sqrt{R^2 + 1}}{R - 1}\right) \ln\left(\frac{1 - X}{1 - R X}\right)}{\ln\left(\frac{\frac{2}{X} - 1 - R + \sqrt{R^2 + 1}}{\frac{2}{X} - 1 - R - \sqrt{R^2 + 1}}\right)}$$

esso serve ad attuare una correzione nella LMTD in ordine del fatto che i flussi dei fluidi non sono esattamente in perfetta controcorrente.

Si riportano quindi le analisi condotte similmente a quelle sugli errori presenti nelle misure delle portate (sezione 5.1 a pagina 39), con gli stessi dati di partenza.

MC	90487.65	2	FC cost	0.9663
MH	22411.95	0.5	FC calc	0.9602

	non corr.	corretta	err %
Th	98.372	98.5193	0.149455
Tc	90.0929	90.086	0.007649

FC corretto si riferisce al calcolo di esso mediante metodo iterativo (si è ricorso all'uso di MATLAB per la soluzione), mentre F costante non prende parte (per la definizione del modello) nel calcolo delle temperature in uscita. Come si può notare l'errore è abbastanza contenuto. Si riportano gli altri casi analizzati.

MC	90487.65	0.5	FC cost	0.9663
MH	22411.95	2	FC calc	0.9668

	non corr.	corretta	err %
Th	141.664	141.655	0.006207
Tc	109.019	109.025	0.006039

MC	90487.65	0.5	FC cost	0.9663
MH	22411.95	0.5	FC calc	0.8684

	non corr.	corretta	E %
Th	100.349	102.956	2.5319
Tc	99.9915	99.5035	0.490421

MC	90487.65	2	FC cost	0.9663
MH	22411.95	2	FC calc	0.9914

	non corr.	corretta	E %
Th	138.716	138.085	0.45738
Tc	92.8091	92.9273	0.12723

Come si può notare l'errore è sempre molto contenuto e leggermente più significativo nel caso di portate ampiamente ridotte rispetto al caso iniziale, ciò dipende sostanzialmente dalla forma della funzione FC.

A.2 Analisi degli effetti dovuti alla variazione dei calori specifici in funzione del cambiamento delle condizioni operative

Le variazioni dei calori specifici in funzione della temperatura sono state fornite dal personale ENI (sono tutte correlazioni lineari). Si è quindi attuata la procedura identica a quella sul parametro FC.

MC	180975.3	2*MCi
MH	11205.98	0.5*MHi

	non coretto	corretto	err %
Th	99.0823	98.9937	0.089514
Tc	90.145	90.1841	0.043455

MC	45243.825	0.5*MCi
MH	44823.9	2*MCi

	non coretto	corretto	err %
Th	142.24	142.303	0.044023
Tc	109.139	109.031	0.098591

MC	45243.825	0.5*MCi
MH	11205.975	0.5*MHi

	non coretto	corretto	err %
Th	103.557	103.622	0.062559
Tc	99.711	99.8584	0.147635

MC	180975.3	2*MCi
MH	44823.9	2*MHi

	non coretto	corretto	err %
Th	138.719	138.622	0.070037
Tc	92.963	92.9291	0.036394

Anche in questo caso si può notare come gli errori siano veramente poco significativi.

A.3 Analisi delle conseguenze della variabilità del coefficiente globale di scambio termico in relazione alle diverse condizioni operative

Essendo difficoltoso valutare le variazioni di U in funzione delle temperature e delle portate a causa della scarsa conoscenza delle proprietà del fluido, si è proceduto ipotizzando per determinate variazioni della portata diverse possibili variazioni di U. Si ipotizza che U cresca in percentuale in modo più contenuto rispetto alle portate. Si sono analizzati i casi peggiori, ovvero di variazione positiva e negativa contemporaneamente per entrambe le portate. A rigor di logica si può ipotizzare che aumentando le portate aumenterà il coefficiente di scambio termico, viceversa diminuirà con una riduzione. Si riportano sinteticamente i dati.

Variazione di U: per portate crescenti

E % Th	1.748	3.45389	5.11679
E % Tc	0.475618	0.92015	1.33597
coef molt su U	1.1	1.2	1.3

per portate decrescenti

E % Th	1.93633	4.24475	6.97435
E % Tc	0.383935	0.866079	1.47359
coef molt	0.9	0.8	0.7

Anche in questo caso si nota come gli errori siano contenuti. Effettivamente per piccole variazioni di portata l'effetto della variazione del coefficiente di scambio termico globale dovrebbe non essere significativo.

A.4 Errori dovuti al calcolo di una temperatura incognita

Nel caso sia assente una temperatura in uscita di uno scambiatore essa viene calcolata dalla legge di conservazione dell'energia:

$$T_H^{out} = -\frac{M_c c_{p,c}}{M_h c_{p,h}} (T_C^{out} - T_C^{in}) + T_H^{in}$$

$$T_C^{out} = \frac{M_c c_{p,c}}{M_h c_{p,h}} (T_H^{in} - T_C^{out}) + T_C^{in}$$

Calcolare le grandezze in questo modo pone il rischio di ottenere dei valori non veritieri. In queste equazioni compaiono varie grandezze fisiche (e non i rapporti ma le grandezze assolute), un errore consistente sulle portate ad esempio (ancora di più se volumetriche), può portare a errori consistenti. Si deve quindi porre attenzione il più possibile nell'evitare di eseguire il programma con troppe temperature incognite.

Riferimenti bibliografici

- [1] Andrea Esposito, Implementazione DMCplus, APC Meeting 2013, Raff. Livorno.
- [2] Fact Book 2012, ENI, http://www.eni.com/en_IT/attachments/publications/reports/reports-2012/fact-book-2012-eng.pdf.
- [3] Aggiornamento della dichiarazione ambientale 2004 Rev . 2 anno 2006, ENI.
- [4] Thompson, D. W. (1987), Fluid mechanics and transfer processes, J. M. Kay and R. M. Nedderman, Cambridge University Press, 602 pages, 1985.
- [5] Saša Singer and John Nelder (2009) Nelder-Mead algorithm. Scholarpedia, 4(7):2928.
- [6] Singer, Saša and Singer, Sanja (2004), “Efficient Implementation of the Nelder-Mead Search Algorithm”, Appl. Numer. Anal. Comput. Math. 1, No. 3, pp. 524–534.
- [7] Singer, Sanja and Singer, Saša (2001), “Complexity Analysis of Nelder-Mead Search Iterations”, in Proceedings of the 1. Conference on Applied Mathematics and Computation, Dubrovnik, Croatia, M. Rogina, V. Hari, N. Limić, and Z. Tutek (Eds.), PMF–Matematički odjel, Zagreb, pp. 185–196. 1999.
- [8] M. A. Luersen and R. Le Riche, Globalized Nelder-Mead method for engineering optimization, in Proceedings of the third international conference on Engineering computational technology (ICECT’03), B. H. V. Topping and Z. Bittnar (Eds.). Civil-Comp press, Edinburgh, UK, UK, 165-166. 2002.
- [9] M. Minoux, "Mathematical Programming: Theory and Algorithm", John Wiley & Sons, 1986.

- [10] R.T. Rockafellar, "Lagrange multipliers in optimization", Nonlinear Programming, Proc. SIAM-AMS, 9, 145-168, R.W. Cottle and C.E. Lemke, eds., New York, 1976.
- [11] T. R. Rockafellar , Convex Analysis. Princeton University Press, 1970.
- [12] Nelder, John A.; R. Mead, "A simplex method for function minimization". Computer Journal 7: 308–313. 1965.