Alma Mater Studiorum · Università di Bologna

facoltà di scienze matematiche, fisiche e naturali
Corso di Laurea Magistrale in Informatica

"Vehicle Routing Problem": un caso di studio

Relatore:
Chiar.mo Prof.
Fabio Panzieri

Correlatore:
Ing. Alberto Torrini

Presentata da:
Davide Malagoli

Sessione II

Anno Accademico 2010-2011

Indice

Elenco delle figure

Elenco delle tabelle

Introduzione

Scopo della tesi

In quest'elaborato si vuole descrivere la mia breve esperienza maturata presso Wincor-Nixdorf, in particolare rispetto ad un problema posto da uno dei suoi clienti, Primafrost.

Primafrost si occupa di servire i numerosi canali di vendita del Gruppo Lombardini (supermercati, discount,ipermercati, cash & carry, e-commerce, grossisti) ponendo sempre molta attenzione non solo alla qualità della merce trasportata, ma anche alla precisione dell'ordine evaso e alla tempestività della distribuzione. I suoi punti di vendita sono sparsi per tutta la parte settentrionale dell'Italia, mentre il magazzino centrale è situato vicino a Mantova.

Attualmente il numero dei punti di vendita è in continuo aumento, e Primafrost doveva rivedere il metodo con cui gestiva la creazione delle gite ed il dispiegamento della sua flotta di veicoli.

Finora infatti tutto questo era stato gestito in modo quasi esclusivamente manuale: lo sforzo principale era infatti svolto da un solo dipendente, il quale in quanto ex-camionista aveva grande esperienza delle tessuto della rete stradale italiana. La sua esperienza era vitale per l'azienda, in quanto permetteva di evitare le strade generalmente più disagevoli per i trasportatori e contemporaneamente di rispettare i vincoli del codice stradale.

Tuttavia ogni infrastruttura che si regge su un solo elemento insostituibile ha ovvi problemi di scalabilità, e la crescente richiesta aveva fatto sì che anche i vertici dell'azienda si rendessero conto del problema, chiedendo a Wincor se era possibile automatizzare, almeno parzialmente, il processo.

Wincor Nixdorf da parte sua aveva già notato che l'interesse dei suoi clienti verso problemi di questo tipo era in crescita, tuttavia tutti quelli che avevano tentato di risolvere il problema da soli con soluzioni personalizzate fino ad allora avevano fallito. Facendo teso di queste esperienze, Wincor aveva deciso quindi di accettare la sfida, tentando però una strada diversa, che passasse attraverso l'integrazione dei componenti che aveva già sviluppato, in modo da offrire ai suoi clienti un servizio che sfruttasse gli anni di esperienza acquisiti nel settore.

Motivazioni

Problemi come questo non sono nuovi nel mondo della Ricerca Operativa: il problema di Primafrost era infatti riconducibile ad un problema piuttosto noto in campo accademico, detto "Vehicle routing Problem". Questo particolare problema è tuttora estesamente trattato in ambito di ricerca, nonostante siano già stati spesi già 40 anni di lavoro. sono state studiate istanze con più o meno

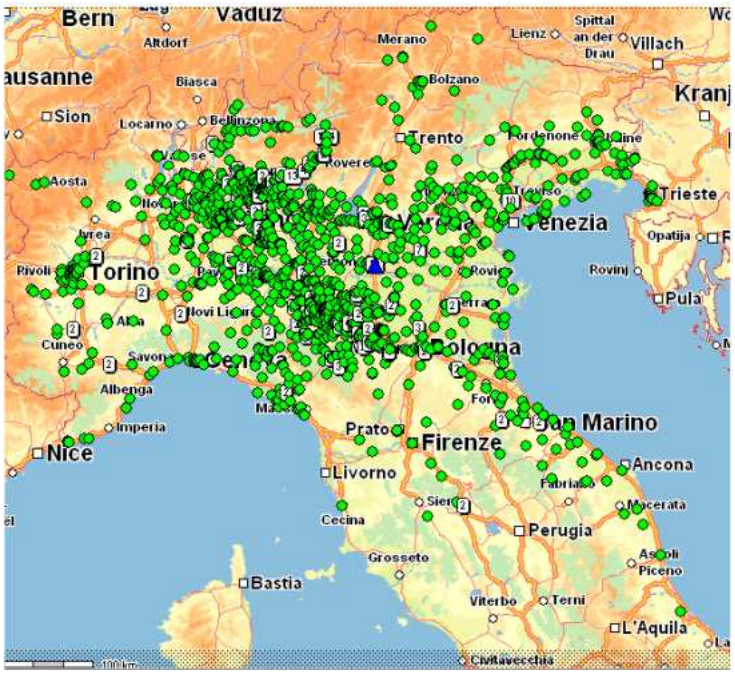


Figura 1: Distribuzione dei clienti di Primafrost nel Nord Italia, aggiornata al 24/09/2010

ogni genere di limitazione ragionevolmente significativa, dalla capacità limitata dei veicoli fino a limitazioni presenti sui punti di consegna o sulle qualifiche dei conducenti.

Questa tipologia di problemi infatti è interessante sia da un punto di vista accademico per la sua elevata complessità, sia da un punto di vista commerciale, poiché le voci di costo interessate sono in genere dell'ordine di milioni di euro, e continuamente crescenti. Anche la più piccola percentuale di risparmio su totali così grandi è quindi sempre significativa per qualsiasi bilancio aziendale.

La soluzione del problema risultava quindi economicamente vantaggiosa per entrambe le aziende, e lasciava intravedere interessanti possibili futuri sviluppi. In particolare Wincor Nixdorf aveva intenzione, nel caso il progetto fosse andato a buon fine, di proporre l'idea anche ad altri suoi clienti.

Stato dell'arte

Utilizzando la sua solita metodologia, Wincor ha deciso di non agire come una software house sviluppando da subito componenti nuovi, ma ha cercato dapprima se esistevano soluzioni già commercializzate.

Per fortuna i frutti di anni di ricerca da parte del mondo accademico possono oggi essere beneficiati anche dal mondo commerciale: esistono infatti alcune soluzioni software che già oggi risolvono diverse tipologie e varianti di questo problema.

Alcune tra le soluzione maggiormente accreditate appartengono ad una compagnia, PTV, di origine tedesca.

Le soluzioni software di PTV sono oggi utilizzate da un ampio pubblico, e anche da diverse aziende più o meno grandi. Si può dire che ovunque vi sia un problema di instradamento di veicoli vi sia virtualmente spazio per l'adozione dei suo prodotti, quindi si può ben immaginare perché abbia avuto una diffusione così endemica.

Qui in Italia uno dei suoi clienti è nient'altro che Poste Italiane,la quale aveva un ovvio problema nell'ottimizzare le spedizioni postali, e ha trovato conveniente utilizzare un software specializzato per trovare una soluzione.

Come si può intuire, PTV ha diverse partnership sparse un po' per tutta Europa. In Italia il partner di riferimento è TPS, con cui Wincor Nixdorf aveva già preso contatti nel periodo in cui arrivai. Avevano già utilizzato infatti un'altra volta una soluzione PTV sempre per Primafrost.

Il pacchetto impiegato era "Maps&Guide", ed era servito poichè aveva capacità simili a Intertour. Una differenza rispetto a Intertour era nella rappresentazione delle mappe: "Maps&Guide" era in grado di rappresentare l'effettivo percorso del veicolo sul manto stradale con una grafica che a prima vista ricordava il ben più famoso Google Maps. Questo strumento era infatti stato integrato all'interno di un'interfaccia piuttosto complessa per la gestione delle consegne. Quest'interfaccia era anch'essa stata realizzata da Tuttavia questo strumento, benché fosse stato già integrato in un'interfaccia p era limitato, in quanto non era in grado

Contributo della tesi

L'azienda infatti stava guardando il mercato alla ricerca di una soluzione che le paresse adatta, in modo da poter effettuare un primo studio di fattibilità, sulla base del quale avrebbe deciso il modus operandi per risolvere il problema del suo cliente.

Il contributo principale di questa tesi riguarda quindi proprio i dati raccolti ed i risultati ottenuti nell'ambito di questo studio di fattibilità. Il lavoro da me svolto presso Wincor Nixdorf è il risultato di un esperienza di affiancamento con un dipendente dell'azienda, Primo Tilli, che già era stato assegnato a questo problema.

Il mio ingresso nell'azienda è avvenuto un po' dopo l'inizio effettivo del progetto, per cui alcune decisioni erano già state prese, ma le sorprese non sono venute a mancare in ogni caso.

Al momento del mio ingresso, si era ormai inquadrato abbastanza nello specifico quali architetture software potevano soddisfare i requisiti impostici dal cliente. Rimaneva però comunque una scelta tra due alternative: vi era la possibilità di implementare tutto utilizzando un'applicazione "vecchio stampo" (PTV Intertour), ed integrarla sia all'interno di una interfaccia precedentemente sviluppata per Primafrost sia con l'ERP aziendale, oppure si poteva anche valutare di seguire lo stesso processo, ma con server applicativi (XTour e serie XServer) già predisposti anche ad un approccio più simile a quello di un'architettura di servizi.

Da entrambe queste due soluzioni ci si aspettava che producessero buoni risultati, e difatti le premesse c'erano, in quanto a prima vista sembravano utilizzare gli stessi algoritmi, anche se magari in modi simili ma non uguali. Non era infatti chiaro da subito se i risultati prodotti da XTour sarebbero stati gli stessi di PTV Intertour o se dovessero essere raffinati prima da qualche

altro componente della famiglia XServer. Dal punto di vista dell'integrazione invece il margine di variabilità era abbastanza ampio: l'interfaccia sviluppata per Primafrost era stata sviluppata utilizzando AJAX, per cui poteva darsi che l'architettura basata su Web Service fosse più facile da integrare.

Il nostro obiettivo era dunque quello di studiare le due architetture, osservare le loro potenzialità, i loro limiti, ed infine scegliere quali delle due si adattava meglio alle nostre esigenze. L'obiettivo che mi era stato assegnato in particolare era quello di studiare il framework MEFISTO ed il suo funzionamento. Questa indagine mi avrebbe portato, anche se allora non lo immaginavo, a conoscere il vasto background di informazioni che sottostava al funzionamento dei software da noi impiegati.

Metodologia usata

Era importante coinvolgere il cliente e contemporaneamente fargli percepire che da parte nostra vi era interesse per il suo problema. Per questo si era optato per un modello di sviluppo incrementale e con milestone stabilite in accordo con le esigenze di tutti.

Quello che si voleva raggiungere era ottenere risultati che potessero risultare credibili e vicini il più possibile a quella che sarebbe risultata essere la realtà di utilizzo. Non è difficile capire il perché di questa esigenza: il cliente aveva bisogno di sapere il prima possibile se la soluzione che si veniva a profilare poteva avere un reale impiego per cui valesse la pena spendere il proprio tempo e denaro. Fortunatamente, contemporaneamente allo sviluppo della soluzione per l'ottimizzazione delle gite, Primafrost e Wincor Nixdorf stavano mettendo a punto un sistema per il monitoraggio dei veicoli, per cui possiamo dire che la base di dati da utilizzata per gli esperimenti avrebbe coinciso con quella degli esperimenti. Non si trattava di dati particolarmente precisi, poiché anche il progetto di monitoraggio tuttavia era ancora allo stadio embrionale, e c'erano ancora diversi problemi nella fase della raccolta dei dati, ma erano sicuramente dati reali, vicini alla realtà di tutti i giorni.

I dati che noi avevamo riguardavano le gite compiute dagli automezzi, e suddivise in base alla giornata in cui erano state effettuate. Da questo insieme sono state alcune giornate "affidabili", i cui dati erano stati già ricontrollati.

Per il resto è stato necessario solo un breve addestramento da parte di TPS nei nostri confronti per quanto riguardava i prodotti utilizzati, dopodiché eravamo già operativi.

Un ulteriore approfondimento

Durante la mia esperienza con Wincor Nixdorf, siamo arrivati ad un punto dove ci siamo resi conto che non erano gli algoritmi ad avere delle falle, anzi quelli ottenevano ottimi risultati, invece non potevamo essere sicuri che i dati riguardanti le mappe ed i limiti dettati dal manto stradale lo fossero.

Questo perchè gli addetti di TPS ci avevano spiegato solo in un secondo momento che i dati che riguardavano la topografia erano sicuramente giusti e controllati, mentre quelli che riguardavano i limiti imposti dal manto stradale (gallerie basse, limiti di velocità, ecc.) potevano non essere perfettamente aggiornati per il suolo italiano.

Si è dunque dovuto limitare quindi l'ottimizzazione complessiva prodotta dallo strumento, in modo che non producesse gite che si sarebbero poi potute rivelare impraticabili. Per fare questo si è deciso di manipolare i fattori di costo della funzione obiettivo che l'applicazione tentava di minimizzare. Questa operazione è stata svolta quasi completamente in modo manuale, facendo previsioni sul comportamento dell'euristica, che tuttavia se analizzata mostrava diverse discontinuità, impedendo un reale studio.

L'apparente approssimatività di questo approccio era tuttavia compensata dalla rapidità con cui ci si riusciva ad avviciniarsi alla soluzione cercata: in pochi giorni eravamo stati in grado non solo di trovare quali relazioni i vari fattori di costo avessero sulle soluzioni generate, ma eravamo anche riusciti a trovare dei coefficienti in grado di produrre i risultati che cercavamo.

Una volta conclusa la mia esperienza con Primafrost ho successivamente deciso di studiare una soluzione migliore ad un problema in cui ci eravamo imbattuti, tentando un approccio basato su algoritmi genetici. Questa parte è interamente descritta in un capitolo a parte.

Questa idea nasceva dalla mia voglia di trovare un modo automatico o semiautomatico per giungere agli stessi risultati a cui eravamo giunti manualmente. Questo da una parte avrebbe probabilmente permesso di perfezionare un metodo utile per un impiego futuro, e dall'altro magari poteva pure permettere di offrire garanzie sulle proprietà della soluzione trovata.

L'idea di base era di effettuare qualche simulazione con un ristretto numero di agenti, il cui corredo genetico sarebbe stato rappresentato da una coppia di coefficienti. Questi coefficienti avrebbero rappresentato i valori per i fattori di costo che avevamo modificato nell'istanza reale.

Questi valori sarebbero stati utlizzati dal software per generare le gite, restituendo il totale dei kilometri percorsi ed il tempo impiegato per percorrerli. La sopravvivenza di ogni agente sarebbe quindi stata dettata dalla vicinanza della soluzione, ottenuta mediante i suoi geni, ad un valore arbitrario. In questo modo risultavano favoriti nella corsa per la sopravvivenza solo gli agenti che generavano soluzioni vicine al valore che cercavamo, per cui sarebbe bastato vedere quale era la popolazione vincente alla fine dell'esecuzione.

Per avere il massimo dell'attinenza tra i risultati delle istanze classiche e quelli del caso reale avevo pensato di utilizzare gli stessi strumenti che avevo utilizzato in Wincor, o qualcosa del genere, tuttavia non è stato possibile a causa della difficoltà, del tutto comprensibile, a farmi consegnare una versione delle mappe da TPS, in modo da poterle utilizzare per gli esperimenti.

Non potendo accedere ai dati originali, e non trovando un modo per far utilizzare ai componenti dei dati creati appositamente da me, ho deciso di tentare la realizzazione di un'implementazione che mimasse le funzioanlità principali dello strumento, ovvero appunto la creazione di rotte.

Siccome non sapevo nulla sull'euristica utilizzata per produrre le gite, ma sapevo che il componente MEFISTO, che veniva utilizzato per la post-ottimizzazione, era basato sulla metaeuristica Granular Tabu Search, ne ho reimplementato la versione originale, e provata sia su istanze già utilizzate dalla letteratura scientifica che sui dati del caso reale in studio.

I risultati sono stati tutt'altro che strabilianti, così sono andato ad esporre i miei dubbi all'ideatore di Granular Tabu Search, Paolo Toth, il quale mi ha confermato i sospetti che già avevo: avevo implementato correttamente la metaeuristica, ma tutti i risultati finali si basavano su una prima soluzione gen-

erata da una euristica rapida, e nella versione da me utilizzata avevo usato un'euristica troppo semplice, non in grado di risolvere brillentemente i problemi che le sottoponevo. In particolare le istanze classiche risultavano ancora fattibili, anche se non si giungeva a risultati strabilianti, mentre l'applicazione non era ingrado di trovare una soluzione per quanto riguardava il caso reale di Primafrost.

Purtroppo non avevo più tempo per reimplementare l'euristica, e così ho dovuto accontentarmi di ciò che avevo.

Nonostante i dati tutto sommato pessimi e le differenze minime tra i valori delle soluzioni migliori rispetto a quelle pessime, i risultati sono stati accettabili: anche usando poche generazioni, e quindi poche iterazioni della procedura, si notava che venivano chiaramente privilegiati gli agenti vincenti.

I risultati ottenuti, quindi, fanno sperare che l'approccio genetico da me utilizzato sia comunque praticabile, e anzi possa diventare auspicabile nel caso si debba un giorno affrontare un problema dello stesso genere, ma più complesso.

A queste conclusioni sono giunto dopo aver paragonato i tempi impiegati dall'algoritmo genetico con un'altro algoritmo, più stupido, che invece provava tutte le soluzioni possibili. Si notava infatti che, benché quest'ultimo garantisse di trovare una soluzione ottima, invece che una semplicemente buona come nel caso genetico, impiegava anche un tempo che era 10 volte maggiore.

Struttura del documento

Questa tesi è strutturata in modo da ricalcare il più possibile quella che è stata la naturale successione degli eventi, in modo che risulti semplice al lettore comprendere ciò che si è fatto. Siccome la problematica affrontata è piuttosto vasta ed articolata, è stata prevista una sezione preliminare in cui tutti i concetti e termini chiave vengono ripresi, in modo da garantire che la lettura di questo documento possa essere già sufficiente per la comprensione delle soluzioni illustrate.

Il documento è suddiviso in 3 capitoli:

1. **Stato dell'arte:** in questo capitolo vengono descritti tutti i concetti importanti, le parole chiave, gli algoritmi che poi saranno menzionati nel documento. Viene anche presentato un approfondimento sul punto di vista delle aziende coinvolte.

2. **Architettura proposta:** in questo capitolo viene presentato un approfondimento sulle architetture proposte, sul lavoro ed i risultati ottenuti per lo studio di fattibilità, e alcune conclusioni che sono state tratte.

3. **Implementazione:** in quest'ultimo capitolo viene presentato un mio contributo personale ad un problema affrontato nel capitolo 2 ma che era rimasto sostanzialmente irrisolto. Spiegando l'idea da cui sono partito, il procedimento da me adottato ed i risultati finali è possibile formulare alcune ipotesi sulla struttura dell'algoritmo sottostante il framework MEFISTO. Inoltre dai risultati si evincono anche alcune potenzialità e difetti della mia implementazione

Al termine delle analisi e delle ipotesi mostrate all'interno dei capitoli, si giunge quindi alle Conclusioni, in cui si discuterà delle criticità che anora oggi esistono nell'affrontare problemi di questa tipologia. Affermazioni ovviamente supportate dall'esperienza diretta da me compiuta.

Capitolo 1

Stato dell'arte

In questo capitolo riporteremo quello che è stato osservato essere oggigiorno lo stato dell'arte per quanto riguarda il problema trattato.

Nella prima sezione analizzeremo il modo di vedere il problema da parte del mondo industriale, in particolare faremo riferimento allo scenario descritto da Wincor Nixdorf.

Nella sezione successiva introdurremo una serie di termini e concetti che saranno utili più avanti in questa tesi, ed infine nell'ultima sezione parleremo un po' del problema dell'instradamento dei veicoli e delle sue varianti.

1.1 Lo scenario di Wincor Nixdorf

In questa sezione parliamo in breve dell'azienda Wincor Nixdorf, della sua struttura organizzativa, per poi concentrarci sul problema che i suoi clienti si erano trovati ad affrontare. Da lì esamineremo brevemente le motivazioni che hanno spinto Wincor ad intraprendere il progetto di cui si parla in questo elaborato e la metodologia con cui hanno deciso di affrontare il problema. Poiché il lavoro descritto in questo documento riguarda un opera concepita presso Wincor Nixdorf Retail Consulting, solo la realtà di questa compagnia sarà descritta nel dettaglio.

1.1.1 Storia ed organizzazione aziendale

Le origini dell'odierna azienda [11] partono nel lontano 1952, nel "Labor für Impulstechnik" del pioniere informatico Heinz Nixdorf. Nel 1968 diventa Nixdorf Computer AG (1968-1990), poi Siemens-Nixdorf Informationssysteme AG (1990-1998), e Siemens-Nixdorf Retail & Banking Systems GmbH (1998-1999). Wincor Nixdorf venne infine fondata nell'ottobre 1999 come compagnia indipendente. Nel 1999 l'organizzazione viene rilevata dalla finanziaria Kohlberg Kravis Roberts insieme al partner Goldman Sachs, e prende il nome di Wincor Nixdorf.

Il suffiso "Wincor", composto da "win" e "core", rappresenta la promessa di profitto e la competenza della compagnia nel settori retail e banking [10].

Oggi Wincor Nixdorf - quotata alla Borsa di Francoforte - conta a livello mondiale oltre 7.000 dipendenti e un fatturato di circa 2 miliardi di Euro. E' presente in oltre 90 paesi, con 31 filiali consociate nazionali. I dati di mercato

fatti registrare nel 2005 indicano che, nei segmenti degli ATM e dei sistemi POS Retail, Wincor Nixdorf è leader di mercato in Germania e al terzo posto a livello mondiale.[12]

In Germania l'azienda è composta da 20 compagnie:

1. Wincor Nixdorf International GmbH
2. Wincor Nixdorf Banking Consulting GmbH (BCON)
3. Wincor Nixdorf Branch Technology GmbH
4. Wincor Nixdorf GmbH Business Administration Center (BAC)
5. Wincor Nixdorf Customer Care GmbH
6. Wincor Nixdorf Services GmbH
7. Wincor Nixdorf Facility GmbH
8. Wincor Nixdorf Facility Services GmbH
9. Wincor Nixdorf property management Ilmenau GmbH & Co.
10. Wincor Nixdorf Logistics GmbH
11. Wincor Nixdorf Lottery Solutions GmbH
12. Wincor Nixdorf Real Estate GmbH & Co.
13. Wincor Nixdorf Retail Consulting GmbH (RCON)
14. Wincor Nixdorf Retail Services GmbH
15. Wincor Nixdorf Security GmbH
16. Wincor Nixdorf Services GmbH
17. Wincor Nixdorf Technology GmbH
18. Wincor Nixdorf Portavis GmbH
19. Prosystems IT GmbH e Bank Consulting AG, di cui Wincor Nixdorf possiede il 51%

In Italia, Wincor Nixdorf è presente dal 1999 con un Gruppo strutturato in tre società:

1. Wincor Nixdorf
2. Wincor Nixdorf Retail
3. Wincor Nixdorf Retail Consulting

che coprono l'intera offerta di consulenza, prodotti e servizi per i settori bancario e di vendita al dettaglio. Wincor Nixdorf Italia conta un organico di circa 180 persone.

1.1.2 Wincor Nixdorf Retail Consulting

1.1.2.1 La missione aziendale

Wincor Nixdorf Retail Consulting (nel seguito WNRC) ha come scopo quello di offrire consulenza su processi del business aziendale alle aziende della Grande Distribuzione Organizzata (nel seguito GDO).

1.1.2.2 La metodologia di lavoro

Dopo aver disegnato i processi con i clienti sorge il problema di come supportarli dal punto di vista dell'Information Tecnology.

WNRC è anche in grado di sviluppare specifiche applicazioni software a supporto dei processi disegnati con i clienti.

L'approccio di WNRC allo sviluppo software non è quello di una *software house* ma quello di offrire ai clienti "soluzioni" ritagliate su misura dei processi da informatizzare.

Nello sviluppo di queste soluzioni WNRC ha avuto l'abilità di integrarle organicamente le une con le altre; oggi costituiscono una suite di nome Merchandise Retail System (acronimo Me.R.Sy, vedi 1.2.1) che in pratica è un ERP per le aziende della GDO.

Le aree funzionali coperte dall'aziende sono molte, in particolare la logistica dei Centri di Distribuzione (nel seguito Cedi).

1.1.3 L'interesse dei clienti al problema

Tipicamente le aziende trattate da WNRC possiedono centinaia di Punti di Vendita (nel seuito Pdv) nei vari canali (Ipermercati, Supermercati, Cash&Carry, Discount, Internet, ecc.)

I Cedi consegnano a questi Pdv diverse volte alla settimana; da cui ne derivano importanti costi di trasporto.

I clienti di Wincor stanno diventando sempre più sensibili a queste voci di costo (anche perchè destinate a crescere nel tempo) e questa area sta diventando per WNRC interessante per nuove possibilità di business.

Negli ultimi anni alcuni clienti di Wincor hanno tentato di affrontare il problema della minimizzazione dei costi di trasporto con dei progetti autonomi, ma sono tutti falliti.

1.1.3.1 Le cause del fallimento dei progetti affrontati dai clienti

Le cause principali del fallimento si ritiene possano essere queste:

1. La complessità del problema è stata sottovalutata

2. L'approccio al problema non è stato quello giusto; ad esempio si è cercato di dare uno strumento sofisiticato in mano al responsabile delle spedizioni il quale:

   (a) non aveva il bagaglio culturale necessario (spesso questa figura è un ex-caminista)

   (b) Provocava un conflitto di interessi (queste figure, se abili, sono ben remunerate; lo strumento, se funziona, erode il potere di queste figure...)

3. Non si è capito la necessità di fortissima integrazione fra lo strumento di integrazione cartografica e quello che, nel processo, sta a monte e a valle (ad esempio, una volta pianificato le gite bisogna essere in grado di pilotare il picking ed il carico del camion in modo da rispettare la pianificazione)

1.1.4 L'interessamento di Wincor al problema

In conclusione Wincor Nixdorf riteneva che:

- I tempi fossero maturi per spingere, con alcuni dei clienti principali, un progetto di "minimizzazione costi trasporto" che, utilizzando un sofisticato tool di painificazione cartografica, permettesse di ridurre questa importante voce di costo.

- condizione necessaria (e non sufficiente) per concludere, assieme ai clienti, il progetto con successo, fosse quella di integrare il tool all'interno di Me.R.Sy., permettendo ai due software di dialogare e di scambiarsi dati

- fosse necessario un approccio propositivo presso i clienti di questo progetto

La fattibilità di questo progetto era tuttavia subordinata al successo di un "Proof of Concept" nel quale si sarebbero dovuti fornire anche indicazioni sull'incidenza del risparmio potenziale.

1.2 Background

In questa sezione è possibile trovare chiarimenti sui termini che verranno usati nelle sezioni successive.

1.2.1 Me.R.Sy

"Me.R.Sy."(Merchandising Retail System) [1] è la suite gestionale di Wincor Nixdorf. risponde alle esigenze delle aziende retail gestendo l'intera filiera della moderna distribuzione: dalla negoziazione col fornitore alla logistica della merce, dal punto di vendita al consumatore finale.

La suite applicativa Me.R.Sy., grazie alla sua modularità, che permette di configurare l'applicazione in modo graduale e progressivo nel tempo, controlla con efficacia ed efficienza tutti i processi caratteristici di un'azienda retail, garantendo alti livelli di flessibilità.

I moduli della Suite Me.R.Sy. insistono su 4 aree applicative:

1. **Area Commerciale:** supporta l'impresa e gli utenti nell'ambito della relazione con l'industria. L'ambiente applicativo, basandosi su questa solida struttura anagrafica, garantisce una copertura ampia e profonda di tutte le funzionalità inerenti il ciclo passivo (rapporto tra distributore e produttore) e il ciclo attivo (rapporto tra distributore e punto vendita).

2. **Area Logistica:** supporta l'impresa e gli utenti nell'ambito della gestione del ciclo merci e del magazzino. Nella gestione dei processi caratteristici, Me.R.Sy. può utilizzare tecnologie sia in ambito radiofrequenza (RF) che in ambito voice, garantendo precisione nell'esecuzione delle attività e una tracciabilità completa dei prodotti.

3. **Gestione della Rete:** permette la gestione delle superfici di vendita evolutasi, nel modello e nelle funzioni, con le best practice italiane del canale. Integra in sé tutte le anagrafiche necessarie alla completa gestione dei negozi, nonché la possibilità di utilizzare, per le politiche commerciali, alcuni specifici moduli per la progettazione, applicazione e controllo della politica commerciale.

4. **Area Store:** garantisce un'ampia copertura funzionale per i processi di back office, mantenendo in loco la sola interfaccia con la barriera casse e le bilance.

1.2.2 Geocodifica

La geocodifica [3] consente di attribuire ad un indirizzo presente in un database (via, numero civico, località, provincia, nazione) una coppia di coordinate geografiche. La geocodifica consente il posizionamento degli indirizzi sulla cartografia digitale e, quindi, la loro successiva visualizzazione e rielaborazione per attività di analisi, di pianificazione, di geomarketing, etc.

1.2.3 Gite e gite cartografiche

Il percorso che un veicolo terrestre (es. automobile,camion, treno, ecc.), navale, o aereo) compie partendo dal punto di partenza a quello di arrivo, e passando nel suo tragitto attraverso vari punti di controllo, non ha una denominazione univoca in italiano: "giro" o "tragitto" sono in genere le parole più appropriate. Nei problemi di instradamento dei veicoli (vedi 1.3) l'insieme dei veicoli potrebbe tranquillamente essere composto anche da veicoli navali o aerei, senza cambiare di molto la tecnica di soluzione del problema stesso. Per questo in questo elaborato si parlerà in modo generico di "gite".

Con il termine "gita" si intende il percorso che un veicolo (terrestre navale, o aereo) compie partendo dal punto di partenza a quello di arrivo, e passando nel suo tragitto attraverso vari punti di controllo, che sono rappresentati dai punti di consegna.

Una gita è quindi una sequenza ordinata di cui ogni elemento rappresenta un punto di consegna. Ogni gita ha come primo elemento sempre un deposito tra quelli che sono stati definiti.

Una gita cartografica è una gita in cui i punti di consegna sono geocodificati.

In questo elaborato ogni volta in cui si parla di gite si intende gite cartografiche.

1.2.3.1 Punti di consegna

Un punto di consegna rappresenta un cliente che fa richiesta ad un deposito di un qualche tipo di bene o merce, e che deve essergli recapitato alle sue coordinate.

Ogni punto di consegna possiede un identificativo univoco ed una serie di coordinate che permette di localizzarlo all'interno dello spazio del problema. Nel caso lo spazio del problema sia rappresentato da mappe geografiche generalmente si utilizzano coordinate geografiche, mentre nel caso di mappe cartografiche digitali si può passare anche a tecniche di geocodifica.

La definizione dell'unità di misura della merce è in genere definita o concordata con l'utilizzatore finale, quindi nel caso di merci possono essere pallet, roll, ecc. o anche confezioni singole.

1.2.3.2 Depositi

Sono punti di consegna speciali, che hanno una richiesta nulla (i depositi non possono fare richieste tra di loro). Ogni gita parte sempre da uno dei depositi. In questo elaborato sarà trattati solamente casi in cui vi è un solo deposito, quindi sarà anche rispettata la convenzione utilizzata in letteratura di indicare il deposito con l'identificativo 0.

1.2.3.3 Gite "aperte" e gite "chiuse"

Una gita, per essere tale secondo la nostra definizione, deve sempre partire dal deposito. Una gita si dice "chiusa" se il punto di arrivo è il deposito di partenza, "aperta" altrimenti.

Nei problemi di instradamento dei veicoli classici normalmente si suppone che i veicoli debbano partire e tornare al deposito di partenza. Questo elaborato non si discosta dalla letteratura da questo punto di vista, quindi quando si parla di gite senza specificare diversamente si suppone che si stia parlando di gite "chiuse".

1.2.4 PTV AG

PTV (Planung Transport Verkehr) AG [14], costituita nel 1979 a Karlsruhe, Germania è una compagnia di consulenza e sviluppo software.

Il business di PTV è principalmente concentrato nelle aree del trasporto, della mobilità e della logistica, e sviluppa software per la pianificazione e l'ottimizzazione dei trasporti e delle gite.

La compagnia è ora attiva in 4 continenti, ha 700 dipendenti e 1500 clienti in 75 nazioni. Alcuni dei loro prodotti software sono utilizzati in Germania anche dalle università nella formazione dei futuri ingegneri del traffico.

Una famiglia di prodotti commerciali che sicuramente ha conquistato una certa fetta di mercato per quanto riguarda la pianificazione delle gite è rappresentata dal marchio "Maps & Guide".

1.2.5 TPS

TPS (Transport Planning Service) [13] nasce a Perugia nel 1993 dall'iniziativa di un giovane gruppo di ingegneri specialisti nel settore della pianificazione dei trasporti. L'obiettivo comune era ed è quello di fornire ad operatori pubblici e privati, consulenze, prodotti e servizi nel campo dell'ingegneria del traffico, dei trasporti e della logistica. In questo ambito, TPS ha avviato un'importantissima partnership con la PTV AG, azienda leader a livello europeo nello sviluppo di soluzioni software per la mobilità.

La struttura organizzativa della TPS è articolata sul territorio nazionale italiano con uffici a Perugia (sede legale ed operativa con certificazione di qualità UNI EN ISO 9001:2000), Bologna e Milano.

A livello operativo TPS è strutturata nelle seguenti unità:

- team di progettazione e consulenza nei campi dell'ingegneria del traffico, della logistica e dei trasporti
- unità tecnica per rilievi ed indagini nei settori del traffico e dei trasporti
- rete commerciale e unità tecnica di supporto per l'assistenza post-vendita dei software
- laboratorio di sviluppo software laboratorio di ricerca

L'organico della TPS è per la quasi totalità costituito da ingegneri ed architetti esperti in pianificazione dei trasporti, ingegneria del traffico e modellistica applicata. Ad essi si affianca l'attività di un nucleo di programmatori che lavora nello sviluppo di soluzioni software basate su cartografia digitale.

1.2.6 Euristica di Clarke e Wright

L'euristica di Clarke e Wright2 di cui si parla in questo elaborato di tesi è meglio conosciuta come "algoritmo dei risparmi di Clarke e Wright" (nome originale: "Clarke and Wright Savings Algorithm").

È senz'altro l'euristica più conosciuta nell'ambito del Vehicle Routing Problem, ed è basato sull'idea di risparmio.

Si suddivide in 2 fasi: il calcolo dei risparmi e la fusione delle gite.

Per quanto concerne il calcolo dei risparmi, quando due gite $(0, \ldots, i, 0)$ e $(0, j, \ldots, 0)$ esiste un cammino tra i e j e la nuova gita creata sarebbe ancora fattibile, allora viene computato un'ammontare chiamato risparmio, che è pari a $s_{ij} = c_{i0} + c_{0j} - c_{ij}$, ovvero la distanza che si è risparmiato tornando al deposito una volta di meno. I risparmi vengono quindi ordinati in modo non crescente.

La fusione delle gite può essere fatta in due modi: sequenziale o parallelo.

Nel caso parallelo, partendo dal massimo risparmio fino al minimo si effettuano le giunzioni delle gite, sempre però cercando di rispettare i vincoli di fattibilità.

Nel caso sequenziale invece si esamina una gita per volta, e si cerca tra i risparmi, sempre dal maggiore al minore, se si può combinare la gita corrente con un altra rispettando i vincoli di fattibilità.

I risultati sperimentali dicono che la versione parallela è più performante di quella sequenziale, e quindi in questo elaborato quando ci si riferirà a questa euristica si intenderà sempre la sua versione parallela.

1.2.7 Euristiche e Metaeuristiche

Sono state proposte diverse approcci basati su euristiche per risolvere il problema dell'instradamento dei veicoli (vedi 1.3), che possono essere raggruppati in due insiemi distinti:

1. Quelli basati su euristiche "classiche" (sviluppate principalmente tra gli anni '60 e inizio '90)

2. Quelli basati su "metaeuristiche" (sviluppate tra la metà degli anni '90 ed i primi anni del terzo millennio)

Le euristiche compiono una esplorazione limitata dell'albero delle scelte e delle combinazioni possibili producendo però generalmente una soluzione comunque buona rispetto a quella ottima. Gran parte delle soluzioni commerciali oggigiorno esistenti appartengono alla prima famiglia, anche e sopratutto perchè questo tipo di euristiche può essere esteso per tenere conto della diversità dei vincoli riscontati in contesti reali.

Vedremo invece come le soluzioni analizzate in quest'elaborato (PTV Intertour Standard e gli xServer) appantegono alla seconda famiglia, rappresentando così un'evoluzione rispetto al panorama generale.

Nelle metaeuristiche infatti l'enfasi è posta invece sull'esplorazione in profondità dell'albero delle scelte, o almeno di quelle regioni dell'albero che appaiano più promettenti. Questi metodi tipicamente combinano tecniche sofisticate per la ricerca dei vicini, la memorizzazione delle strutture dati e la ricombinazione dei risultati.

La qualità delle soluzioni così ottenute è più alta rispetto a quella delle euristiche più classiche, ma richiede un maggior tempo per essere trovata. Tra l'altro, queste procedure richiedono una configurazione abbastanza precisa di alcuni parametri utilizzati dall'algoritmo, e che servono all'eurisitica per adattarsi al contesto di impiego. Questa configurazione in alcuni casi può diventare particolarmente problematica.

Nonostante tutto però la differenza tra euristiche e metaeuristiche sembra risiedere nel fatto che le seconde posseggono una serie di accorgimenti per evitare di rimanere bloccati in minimi locali mediante una gestione più permissiva del risultato trovato: nelle euristiche classiche spesso si passa infatti da una soluzione fattibile all'altra, in una continua ricerca di miglioramento; nelle metaeuristiche spesso invece la scelta della soluzione successiva può riguardare anche soluzioni non fattibili, poichè si è visto che sperimentalmente questo approccio permette di trovare più velocemente la soluzione giusta.

1.2.8 Tabu Search

Tabu Search è una metaeuristica molto famosa all'interno della'mbietne della Ricerca Operativa, e diverse sue estensioni sono state usate anche in ambito VRP.

In Tabu Search vengono analizzate sequenze di soluzioni in modo iterativo: ad ogni iterazione vengono prodotte diverse soluzioni da una variazione della soluzione ottenuta al passo precedente. Da questo insieme viene scelta la configurazione "migliore", ovvero quella che porta più vicino all'obiettivo che si vuole raggiungere.

Per evitare la creazione di cicli nella scelta delle soluzioni, le soluzioni che sono state esaminate recentemente sono contrassegnate come "tabù", il che significa che per un certo numero di iterazioni non possono essere più prese in considerazione.

L'approccio basilare di questa metaeuristica, nell'arco di una decina d'anni, è stato utilizzato da diversi autori, i quali lo hanno in genere arricchito di diverse caratteristiche e migliorie. Ad esempio, per risparmiare memoria, alcune

estensioni di Tabu Search non memorizzano le soluzioni interamente, ma solo qualche attributo.

Tra gli approcci più famosi o innovativi ricordiamo:

- TabuRoute, descritto poco più avanti in questo documento (cap. 1.2.10). Concepito da Gendreau, Laporte e Hertz, integra al suo interno l'euristica GENIUS (vedi 1.2.9).
- Adaptive Memory, di Rochard e Taillard, poichè intriduce il concetto di memoria adattiva per potenziare glia pprocci basati su Tabu Search
- Granular Tabu Search, di Toth e Vigo, poichè tuttora viene ritenuto un ottimo approccio per risolvere l'instradamento dei veicoli

1.2.9 GENIUS

GENIUS [9] è un'euristica utilizzata per risolvere il problema del commesso viaggiatore, ma che è stata impiegata anche in campo VRP da TabuRoute (vedi cap.1.2.10).

L'acronimo in realtà descrive due euristiche, "GENeralized Insertion" e "Ustring and Stringing", e che servono ognuna per un passo specifico di questo algoritmo a due fasi.

La prima si occupa, dato un nodo, di inserirlo all'interno di una sequenza circolare (che di norma è una gita).

La seconda euristica, in un secondo momento, si occupa di migliorare la gita trovata, e funge perciò da port-ottimizzazione.

Gli schemi con cui queste euristiche aggiungono o tolgono nodi dalla gita concedono determinate caratteristiche all'algoritmo, e quindi vengono analizzate di seguito.

1.2.9.1 GENI

"GENeralized Insertion" viene utilizzata per inizializzare la gita.

L'inserzione non viene fatta in maniera casuale ma segue degli schemi, che andiamo ad illustrare:

1. Nel primo schema di inserzione si suppone che il nodo n debba essere inserito tra n_i ed n_j (diversi tra loro), e che esista un nodo n_k diverso dagli altri due. Non è necessario che i due nodi siano adiacenti, come si vede bene nella figura 1.1. Come si vede alcuni segmenti della sequenza sono stati "invertiti". Notare che questo schema si riduce ad un inserzione classica nel caso in cui $n_i = n_{i+1}, n_j = n_{j+1}, n_k = n_{k+1}$.
2. Il secondo schema assomiglia al primo, ma questa volta si suppone anche l'esistenza di un nodo $n_l$diverso dagli altri. notare che in questo caso le "inversioni" nella gita sono numerose, come si vede in figura 1.2.

Degno di nota è il fatto che, per come lavora quest'euristica, non si può partire da una gita vuota, ma bensì devono esserci almeno 3 nodi. Questo problema è dovuto alla struttura degli schemi di inserzione, ed è possibile risolverlo aggiungendo nodi casulamente alla gita fino ad arrivare a 3. Un'altra caratteristica che deriva dall'uso di questi schemi è la non necessità di un componente che perturbi la soluzione, poichè le "inversioni" negli schemi servono proprio a questo.

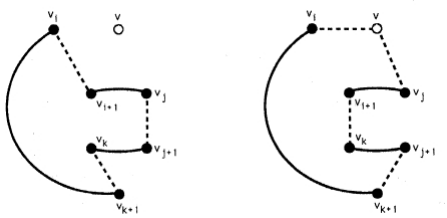


Figura 1.1: Inserzione di tipo 1

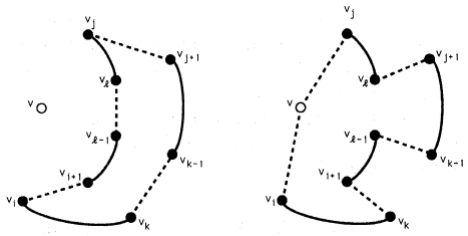


Figura 1.2: Inserzione di tipo 2

Visto che esitono 2 schemi di inserzione, viene sempre scelto l'inserimento che porta alla costruzione di una gita di costo minimo tra le alternative.

1.2.9.2 US

In modo iterativo "Unstringing and Stringing", toglie un nodo dalla gita e lo reiserisce, sempre usando gli schemi di GENI per il reinserimento. La rimozione dei nodi dalla gita non viene fatto a caso, ma seguendo determinati schemi, che come si può notare dalla figure 1.3 e 1.4, altro non sono che l'inserso dei passi di inserzione descritti nel paragrafo precedente.

1.2.10 TabuRoute

Tra gli approcci derivati da Tabu Search, TabuRoute [1.2.10] introduce diverse caratteristiche innovative. La lista delle soluzioni che possono essere raggiunte da quella corrente viene determinata usando GENI (vedi 1.2.9.1), eliminando un vertice da una gita e cercando di inserirlo in un'altra mediante un criterio di minimo costo.

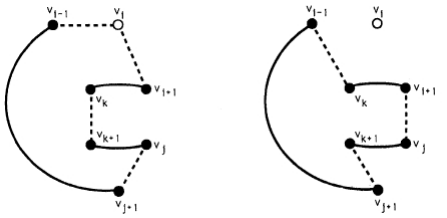


Figura 1.3: Rimozione di tipo 1

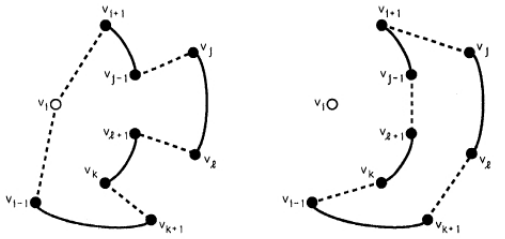


Figura 1.4: Rimozione di tipo 2

Un'altra importante caratteristica consiste nell'utilizzo di fattori di penalizzazione per svantaggiare la scelta di soluzioni che non rispettano i vincoli (dette anche inammissibili). Più precisamente:

- la funzione di ricerca viene lasciata libera di selezionare anche soluzioni inammissibili, a patto che abbiano un costo minore.
- Dopo un certo numero numero di soluzioni inammissibili esaminate consecutivamente i fattori di penalità aumentano, e così aumenta anche il costo delle soluzioni che violano i vincoli
- Dopo un certo numero di soluzioni ammissibili consecutive invece il valore dei fattori di penalità decresce

Questa tecnica viene utilizzata per creare un mix di soluzioni ammissibili e non che impedisca alla ricerca di fermarsi in un minimo locale.

Ultima importante peculiarità di questo algoritmo è l'utilizzo di false partenze: più soluzioni iniziali vengono generate, e solo la più promettente viene scelta come putno di partenza.

1.2.11 Granular Tabu Search

È una metaeuristica, introdotta da Toth e Vigo, che ha finora ottenuto eccellenti risultati per il VRP [8].

L'idea di base, molto promettente, e che gli archi lunghi difficilmente faranno parte della soluzione ottima. Viene quindi introdotto un limite "di granularità" per gli archi, scelto anche in base alla lunghezza media degli archi contenuti in una soluzione ottenuta in un'euristica veloce (nei risultati sperimentali era stata usata quella di Clarke e Wright).

L'implementazione di Granular Tabu Search quindi riprende quella di TabuRoute e vi aggiunge quest'ultima caratteristica, diminuendo così i tempi di calcolo.

Nella versione originale sono considerate solo mosse di tipo OR1 o OR2 per lo scambio dei nodi all'interno delle gite.

1.2.12 PTV Intertour

PTV Intertour non è un'applicativo, ma bensì una suite di programmi specializzati per la pianificazione e l'ottimizzazione dei giri per la grande distribuzione [5]. Ogni software ha alcune funzionalità leggermente diverse che lo rende adatto ad uno specifico compito.

Funzionalità	Prodotto
Pianificazione automatica dei giri di distribuzione	PTV Intertour Standard
Master route planning	PTV Intertour Strategy
Location planning	PTV Intertour Strategy
Pianificazione delle zone di vendita e ottimizzazione della rete commerciale	PTV Map&Market
Componenti server per l'integrazione di funzionalità geografiche e di ottimizzazione dei percorsi in applicativi distribuiti	PTV xServer

Qui di seguito approfondiremo solo gli applicativi che sono interessanti per questo elaborato.

1.2.12.1 PTV Intertour Standard

PTV Intertour è un software che permette di risolvere complessi scenari di distribuzione e raccolta merci sul territorio. Suggerendo una programmazione ottimizzata dei viaggi e degli impieghi della flotta veicolare, sia su base giornaliera che settimanale, consente di risparmiare fino al 20% dei costi di distribuzione. Per garantire risultati di questo tipo Intertour include un framework per l'ottimizzazione (MEFISTO, vedi pag. 14) e ed una dettagliata cartografia digitale su cui viene effettuato il calcolo delle distanze stradali e dei tempi di percorrenza tra le destinazioni. PTV Intertour può generare automaticamente i viaggi assegnando fermate e mezzi in pochi secondi, oltre a permettere anche di aggiungere le fermate una ad una senza compromettere l'efficienza dei viaggi nel complesso. E' possibile tuttavia manipolare la soluzione trovata per adattarla meglio alle proprie esigenze, aggiungendo,togliendo o scambiando fermate tra i viaggi, o addirittura aggiungere nuove fermate non precedentemente pianificate all'interno di un giro. Nella composizione dei viaggi PTV Intertour tiene conto di tutti i possibili vincoli sia temporali, sia strutturali che logistici: orari di servizio per la consegna/ritiro con più fasce temporali giornaliere; tempi di turnazione dei mezzi; data e orari di disponibilità della merce; tempi di guida nel rispetto dei limiti di legge; tempo massimo di permanenza a bordo della merce; quantitativi da consegnare (roll, pallet, volume, peso); capacità delle diverse tipologie di mezzi della flotta; richiesta di equipaggiamenti particolari sui mezzi per determinate merci, clienti; depositi di distribuzione/transit point/multidrop; vincoli di passaggio intermedio su magazzini per il prelievo/scarico della merce; limitazione della durata dei viaggi; generazione di viaggi con concarico.

1.2.12.2 PTV xServer

I PTV xServer basilarmente offrono le stesse funzionalità dei componenti della suite Intertour, ma usando un'architettura orientata ai servizi. Questo comporta una serie di indiscutibili vantaggi:

1. permettono di ideare soluzioni con un'occhio di riguardo per la scalabilità. Gli xServer sono indipendenti l'uno dall'altro, e grazie alla tecnologia clus-

ter, si può espandere il pool di macchine utilizzate in qualsiasi momento e quindi soddisfare eventuali richieste di prestazioni maggiori

2. sono compatibili con le tecnologie attuali di sviluppo, tutti i componenti hanno interfaccia standard XML/SOAP e questo significa che possono essere facilmente integrati in sistemi ed applicativi

3. i bundle degli xServer sono completamente standalone, e non necessitano di nessuna installazione

4. mantengono performance elevate poichè l'integrazione dei moduli avviene sì tramite un framework di comunicazione scritto in Java ma i singoli moduli sono scritti in linguaggio C++

5. sono multipiattaforma, in quanto possono essere installati in ambiente Windows o Linux e sono in grado di sfruttare i sistemi multiprocessore

6. hanno un approccio nativamente Web 2.0, con interattività garantita grazie all'utilizzo della tecnologia AJAX

7. vi è un intero portale completamente dedicato al prodotto ("PTV Developer Zone") dove consultare tutte le ultime novità sui prodotti disponibili, tutte le statistiche sul materiale cartografico, scaricare aggiornamenti e nuove release, consultare FAQ ed ottenere supporto

Qui di seguito riportiamo i 7 server, ognuno con il compito ad esso associato:

1. **xMap:** si occupa della navigazione dinamica delle mappe AJAX e rappresentazione interattiva di percorsi, informazioni sul traffico, veicoli e, più in generale, POI, punti, polilinee, aree, immagini, etc.

2. **xLocate:** si occupa della geocodifica e validazione di indirizzi (dall'indirizzo alle coordinate). Supporta differenti formati di coordinate geografiche (WGS-84, GeoDecimal, GeoMinSec, Mercator etc.), permette la geocodifica inversa (dalle coordinate all'indirizzo). Possiede un modalità batch per la geocodifica di grandi quantità di indirizzi e differenti metodi di ricerca che possono essere combinati l'uno con l'altro (binary, fuzzy, phonetic)

3. **xRoute:** possiede tutte le funzionalità legate al calcolo di percorsi con più tappe intermedie, distanze e tempi di percorrenza (generazione della lista descrittiva del percorso e dei dati necessari alla rappresentazione del percorso su mappa). Permette inoltre di calcolare i percorsi per diversi tipologie di veicolo tenendo conto di pedaggi autostradali.

4. **xSequence:** ottimizzazione di una sequenza di tappe nel rispetto di vincoli operativi quali finestre orarie e molti altri parametri propri del mondo del trasporto e della logistica (carico/scarico, quantità, categorie di veicoli,...). La differenza principale rispetto a xRoute sta nel fatto che viene considerata una sola sequenza, non tutti i percorsi contemporaneamente

5. **xDima:** calcolo ottimizzato di matrici di distanza (tempi e distanze).

6. **xCluster:** utile per la creazione di gruppi omogenei di punti in funzione di parametri geografici e altri criteri.

7. **xTour:** ottimizzazione dei giri in funzione di parametri quali la capacità dei veicoli, le finestre temporali di aperture e chiusura, etc. Gli ordini vengono suddivisi in giri ottimizzati nel rispetto dei vincoli riducendo i chilometri percorsi, il tempo e il numero di veicoli necessari. La differenza principale ripetto a xCluster sta nel fatto che xTour non si occupa di raggruppare i clienti in base alle consegne periodiche, mentre la differenza rispetto a xRoute è rappresentata dalla strategia di ottimizzazione, in particolare xTour tiene conto anche delle finestre di consegna.

1.2.13 MEFISTO

MEFISTO (Metaheuristics Framework for Information Systems in Transport Optimisation) [6] è, come dice il nome, un framework per l'implementazione di metaeuristiche, disegnato dai suoi autori ed ora sfruttato da PTV per le sue soluzioni commerciali.

Le sviluppo di MEFISTO è dovuto al fatto che fino a poco tempo fa i requisiti per un componente business in ambito VRP erano:

- la correttezza del modello, che doveva essere più vicino alla realtà possibile
- il tempo di calcolo, che doveva essere il minimo possibile o comunque un ammontare ragionevole

Questo aveva portato a vincoli più stringenti come distanze asimmetriche, finestre temporali diverse per ogni utente, tempi di attesa massimi, eterogneità della flotta, vincoli di precedenza, ecc, mentre i tempi di calcolo venivano ridotti affidandosi ad eursitiche particolarmente rapide e ad una successiva fase di post-ottimizzazione (come succedeva nel caso di GENIUS, vedi 1.2.9).

Questa classe di problemi, in letteratura conosciuta come 'problemi "ricchi" di instradamento dei veicoli' (nome originale: "rich veihicle routing problems"), ha avuto particolare fortuna negli anni passati.

Recentemente però ha cominciato ad emergere un alteriore requisito, ovvero la qualità della soluzione stessa, richiesta che si aggiungeva ai precedenti vincoli posti. Contemporaneamente l'avanzamento tecnologico dell'hardware aveva permesso di ottenere dei tempi di calcolo sufficientemente brevi a fronte di una buona qualità della soluzione. Questi fattori portarono quindi ad un cambio di atteggiamento verso il problema, e permesso ad approcci più sofisticati, come le metaeuristiche, di poter essere usate su ampia scala, e non solo per soddisfare i bisogni del singolo cliente.

Di metaeuristiche però non ve n'era una sola, e la ricerca in questo campo intanto continuava insesorabilmente. Per mostrare un comportamento trasparente rispetto alla particolare metauristica utilizzata e mantenere compatibilità sulle interfacce, era necessario sviluppare una framework più generale. Da questa idea nacque MEFISTO.

L'intero framework è scritto in C++ e si compone di un'architettura suddivisa in 3 livelli:

1. il primo livello è sostanzialmente indipendente dall'applicazione e dalla metaeuristica implementata. Fornisce soltanto le funzioni principali e le strutture dati che servono a rappresentare le entità fondamentali di qualsiasi metaeuristica. Fanno parte di questo livello le classi che rappresentano

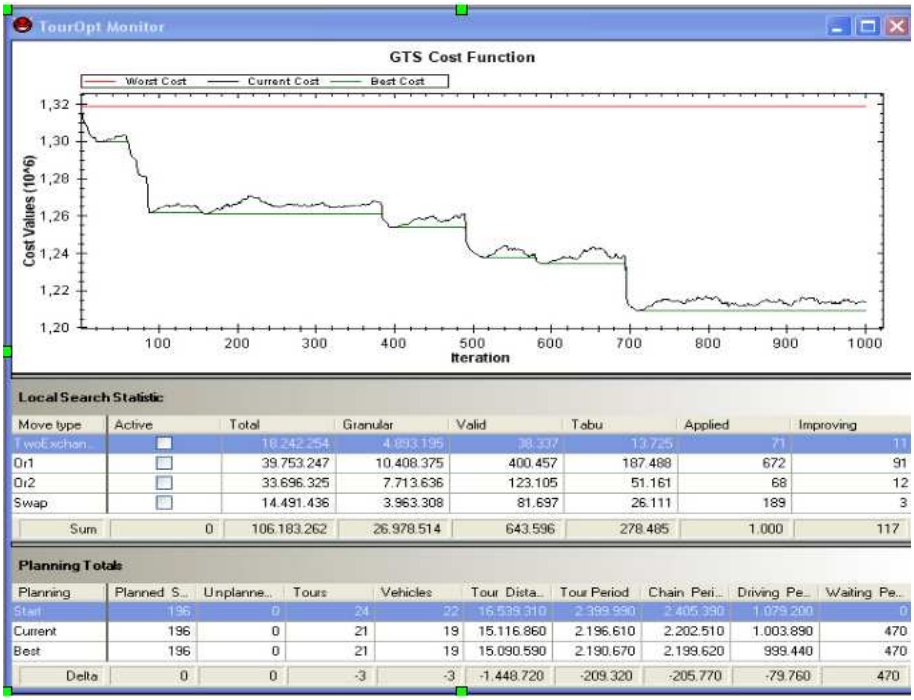


Figura 1.5: *Finestra grafica di MEFISTO*

Si possono notare vari putni di interesse: nella parte superiore, possiamo vedere un grafico che ci illustra quale sia il costo dell'attuale soluzione migliore (linea verde), della soluzione di partenza (linea rossa), e la soluzione corrente (linea nera).
Nella parte inferiore invece possiamo notare più o meno le stesse informazioni, ma in valore numerico.

il generatore di mosse, le mosse stesse, lo spazio delle soluzioni e la ricerca locale

2. il secondo livello contiene tutte le classi che servono alla specifica metaeuristica, ed ereditano le funzionalità dalle classi del primo livello

3. il terzo livello è fortemente dipendente dall'applicazione utilizzata e dal tipo di problema trattato. Ad esempio, per quanto riguarda l'ambito VRP, questo livello dovrà contenere le classi rappresentanti le mosse per scambiare nodi ed archi

Nell'articolo originale vengono inoltre riportati i risultati sperimentali ottenuti in un caso reale e utilizzando un'estensione di Granular Tabu Search, a cui sono state aggiunte anche mosse di swap e A-B.

Dalle informazioni che ci sono state fornite da TPS e PTV questa metaeuristica doveva essere molto simile a quella utilizzata da PTV Intertour Standard e dagli xServer.

Un esempio di funzionamento di MEFISTO mostrato in grafica lo si può vedere in figura 1.5.

1.3 Vehicle Routing Problem

Il problema dell'instradamento dei veicoli (in inglese "Vehicle Routing Problem", o VRP) riguarda la determinazione di un insieme ottimale di gite che dovranno essere utilizzati da una flotta di veicoli per servire un dato insieme di clienti. Come è possibile intuire data la sua similitudine con il problema del commesso viaggiatore, la sua complessità è elevata, apprteiene infatti all'insieme dei problemi NP-hard.

Come problema combinatorio e di ottimizzazione è un classico nella letteratura della Ricerca Operativa: sono passati ormai più di 50 anni da quando Dantzig e Ramser introdussero il problema nel 1959.

Nella letteratura vi è una predisposizione a indicare questo problema con il suo nome od il suo acronimo inglese. Tale convenzione verrà d'ora in poi rispettata anche in questo elaborato.

Il "Vehicle Routing Problem" è un problema importante per il trasporto, la logistica e la distribuzione [4]. Lo scopo implicito è quello di minimizzare il costo per la distribuzione delle merci, quindi è chiaro che vi è anche molto interesse industriale, non solo di ricerca, su questo problema.

A partire dall'articolo di Dantiz e Ramster e successivamente quello di Clarke e Wright (nel 1964), furono quindi concepiti centinaia di modelli e algoritmi furono proposti per trovare una soluzione il più possibile ottima, ma data l'elevata complessità del problema stesso finora sono state prodotte solo euristiche che producono approssimazioni più o meno precise in un tempo ragionevole.

La versione classica del VRP prevede che l'intera flotta parta da un unico deposito, visiti una serie di punti di consegna, e poi ritorni al deposito di partenza. Inoltre tutte le richieste dei clienti sono deterministiche, conosciute in anticipo, e non possono essere suddivise tra i mezzi. Esistono però alcune variazioni, più o meno note, di grande interesse pratico:

- **Capacitated VRP (CVRP)**: in questa versione i mezzi che compongono la flotta hanno una capacità limitata ma uguale per tutti, e che quindi funge da vincolo nel calcolo della soluzione
- **Multi-Capacitated VRP (MCVRP):** come nel CVRP i mezzi hanno una capacità limitata, ma le capacità dei singoli veicoli possono essere diverse tra loro
- **VRP with Time Window (VRPTW):** valgono tutti vincoli descritti per il CVRP, e si aggiunge un vincolo sull'intervallo di tempo in cui la merce deve essere consegnata. Ogni cliente espone un intervallo di tempo in cui è "aperto", e quindi ricettivo alla consegna della merce, chiamata finestra di consegna. Beni consegnati al di fuori della finestra di consegna non vengono raccolti e contano come consegna mancata. Il problema quindi si arrichisce di complessità perchè non è più importante coprire solamente tutti i clienti con un dispendio minimo, ma bisogna anche assicurarsi di consegnare la merce in tempo, in modo da minimizzare l'ammontare di merce non consegnata.

Per ognuna di queste è prevista sia la versione "simmetrica", ovvero dove gli archi hanno lo stesso costo sia in un verso che nell'altro, che quella "asimmetrica", dove invece il costo dell'arco dipende anche dal verso in cui lo si percorre. In questo

elaborato, a meno che non specificato esplicitamente, si intendono sempre le versioni asimmetriche di tali problemi, in quanto le versioni simmetriche non sono che un caso particolare di quelle asimmetriche.

Esistono anche variazioni che contemplano l'utilizzo di più depositi invece di averne uno solo centralizzato, ma non saranno descritte in questo elaborato.

1.4 Algoritmo genetico

Un algoritmo genetico è un'euristica di ricerca che cerca di riprodurre il processo di evoluzione naturale. Quesa euristica è generalemnte usata per genrare utili soluzioni nei problemi di ottimizzazione e ricerca. Gli algoritmi genetici alla classe più generale dei problemi evoluzionistici, i quali generano soluzioni a problemi di ottimizzazione usando tecniche ispirate sempre all'evoluzione naturale, quali ereditarietà, mutazione, selezione e crossover.

1.4.1 Principio generale

In un algoritmo genetico [17], una popolazione di stringhe (chiamate cromosomi, geni o genotipi del genoma) che rappresentano le soluzioni candidate (chiamate individui o fenotipi) in un problema di ottimizzazione, evolvono verso soluzioni migliori.

Tradizionalmente, le soluzioni sono rappresentate come stringhe bnarie, ma sono possibili anche altre rappresentazioni ovviamente. Gli algoritmi genetici sono impiegati in molti campi quali (bio)informatica, ingegneria, economia, chimica, matematica, fisica, ecc.

Un tipico algoritmo genetico richiede:

1. un modo per rappresentare mediante geni il dominio delle soluzioni
2. una funzione detta "di idoneità", per valutare il dominio delle soluzioni

La funzione "di idoneità" viene geeralemnte utilizzata per favorire le soluzioni "idonee" o comunque buone rispetto al problema, e svantaggiare le altre.

Pseudocodice di un semplice algoritmo genetico generazionale:

1. Scegli la popolazione iniziale di individui
2. Valuta l'idoneità di ogni individuo
3. Ripeti fino a che il criterio di terminazione non è raggiunto: (time out, idoneità minima raggiunta, ecc.)
   (a) segli gli individui più idonei per la riproduzione
   (b) dai origine ai nuovi individui attraverso il crossover dei geni e la mutazione
   (c) Valuta l'idoneità di ogni nuovo individuo
   (d) rimpiazza gli individui meno idonei con i nuovi

1.4.2 Inizializzazione

Inizialmente vengono generate tanti corredi genetici quanti individui della popolazione initiale. queste rappresentano le soluzioni iniziali. La dimensione della popolazione iniziale dipende dal problema in questione, ma di solito contiene centinaia o migliaia di individui.

Sempre parlando in linea generale, questa popolazione viene generata casualmente, cercando di coprire l'intero spettro della soluzioni possibili. Oppure si può anche utilizzare particolari intervalli di valori in cui si pensa possano annidarsi le soluzioni migliori.

1.4.3 Selezione

Durante ogni iterazione successiva, una porzione della popolazione esistente viene selezionata per dare vita ad una nuova generazione.

Le soluzioni individuali vengono selzionate attraverso un processo basato sull'idoneità del singolo individuo, dove le soluzioni con un più alto punteggio di idoneità hanno maggiore probabilità di essere selezionate.

Esistono più stategie di scelta, ognuna con i suoi pro e contro:

1. **metodo della roulette:** con questo metodo ogni individuo ha una idoneità f_i ed una probabilità di essere scelto pari a $p_i = \frac{f_i}{\sum_{j=1}^{N} f_j}$, dove N è il numero degli individui scelti. Questo permette di ottenere una selezione abbastanza uniforme e rappresentativa rispetto alla effettiva idoneità degli individui.

2. **campionamento universale stocastico:** simile al metodo della roulette, ma con migliori probabilità matematiche dal punto di vista della previsione

3. **metodo del torneo:** richiede la simulazione di diversi "tornei" tra una ristretta cerchia di individui scelti a caso tra la popolazione. Il vincitore di ogni torneo, ovvero quello più idoneo, viene scelto per la riproduzione. La pressione selettiva indotta da questo metodo può essere moderata variando la dimensione degli individui coinvolti. Più il numero degli individui è grande, minore è la probabilità che gli individui deboli vengano selzionati.

4. **metodo della troncatura:** semplicissimo, semplicemente seleziona i migliori K individui sulla base dell'idoneità e li fa riprodurre tra loro.

1.4.4 Riproduzione

Nel passo successivo viene creata una seconda generazione di individui (o soluzioni) mediante una simulazione della riproduzione sessuale: il crossover.

Per ogni nuova soluzione che si vole produrre, una coppia di soluzioni "genitrici" viene selzionata. Producendo un "bambino" attraverso le operazioni che spiegheremo a breve, viene creata una nuova soluzione che ha molto in comune con i suoi "genitori".

Lo scopo ultimo è quello di far progredire le soluzioni verso una nuova genrazione, potenzialmente "migliore" di quella precedente.

Esistono diversi meccanismi di crossover, alcuni tra i più famosi sono:

1. Crossover puntuale: viene selezionato un indice particolare all'interno delle due stringhe che rappresentano i corredi genetici dei "genitori", e quindi si effettua lo scambio dei due segmenti

2. Crossover a doppio punto: simile al crossover puntuale, ma con 2 punti di scambio

3. "Cut and Splice": simile al crossover puntale, ma i punti di scambio non hanno lo stesso indice. Atipico rispetto ai precedenti poichè può produrre "prole" con stringhe di lunghezza diversa rispetto ai "genitori"

4. Crossover uniforme: si seleziona una probabilità con cui due bit delle due stringhe si scambieranno (tipicamente 0.5) e si lancia una monetina per ogni bit della stringa

All'interno di questo processo la mutazione serve unicamente ad impedire che le soluzioni si adagino su un "minimo locale": se la mutazione è vantaggiosa infatti, l'intero gruppo tende a compiere un balzo in avanti da un punto di vista evoluzionistico, altrimenti viene semplicemente scartata senza alterare il risultato finale.

Ovviamente esistono anche meccanismi diversi dalla mutazione e dal crossover, ma non saranno affrontati in quest'elaborato.

Capitolo 2

Architettura proposta

Il problema posto da Primafrost, che riguardava il carico, scarico e la definizone delle gite per la flotta degli automezzi era facilmente riconducibile ad una particolare istanza di VRP.

Questo tipo di problemi sono tutti NP-hard, tuttavia sono già disponibili diverse soluzioni commerciali (basate su euristiche) che permettono di ottenere soluzioni molto buone, se non ottime.

Quando il sottoscritto ha cominciato ad occuparsi di questo progetto lo studio di fattibilità non era nella sua fase iniziale, per cui alcune scelte erano già state fatte, in quanto si era già arrivati ad una bbozza generale del progetto, e visto un paio di alternative possibili. Si è tuttavia deciso di riportare comunque tutti i dati, sia quelli precedenti al mio arrivo sia quelli successivi per dare un'idea più articolata del contesto di lavoro. Ogni sezione deve essere vista come una fase dello studio di fattibilità,ed alla fine di ogni sezione viene riportato quello che è stato il risultato dell'incontro finale con gli stakeholder.

Tutte le scelte precedenti al mio arrivo verranno presentate nella prima sezione di questo capitolo, e successivamente si passerà al problema vero e proprio: la scelta tra un'architettura "vecchio stampo", con l'utilizzo di software stand-alone, oppure l'adozione di una più moderna architettura Web service ma un attimo più complessa da progettare e realizzare.

2.1 Progetto di massima della soluzione

Precedentemente alla proposta di Wincor Nixdorf, presso Primafrost la gestione della flotta (ovvero carico e gite) era affidata ad un dipendente dell'azienda, il quale essendo un ex-camionista ed avendo una larga esperienza della rete stradale italiana poteva senza dubbio soddisfare le esigenze dell'azienda.

Ma questa gestione portava diversi problemi:

1. Il futuro di quella parte dell'azienda era accentrato tutto nelle mani di un solo uomo

2. La gestione manuale non dava nessuna garanzia di portare a soluzioni ottime o quantomeno buone, benchè si fossero presi diversi accorgimenti e vi fosse buona volontà da parte di tutti

3. Questa gestione non conveniva nemmeno al suddetto dipendente, il quale a caus del suo ruolo poteva prendersi pochissime ferie

Si cercava dunque una soluzione che permettesse di automatizzare l'intero processo, o almeno ua sua parte, in modo da alleviare gli svantaggi dovuti alla situazione attuale.

2.1.1 Requisiti della soluzione

La soluzione proposta, per risultare accettabile da parte di Primafrost, doveva:

1. permettere di automatizzare, in tutto o in parte, la generazione delle gite, una volta forniti ovviamente i dati necessari (mezzi, richieste, ecc.)
2. offrire garanzie di minimizzazione dei costi maggiori rispetto alla normale gestione manuale
3. permettere di modificare le gite generate, possibilmente in maniera semplice e veloce, garantendo così un maggior controllo sull'output prodotto
4. offrire garanzie di rispettare le norme sindacali vigenti per gli autisti dei camion, tra cui è contemplata anche ilrispetto dei tempi di riposo
5. essere pronta, almeno in una forma prototipale, in un tempo relativamente breve

Per essere accettabile da parte di Wincor la soluzione doveva:

1. rispettare i vincoli posti dal cliente
2. permettere una gestione dei dati integrata con l'ERP aziendale, Me.R.Sy

Requisiti impliciti da parte del cliente potevano essere sulla possibile interfaccia del prodotto, che doveva essere user-friendly e non intralciante dal punto di vista operativo.

Da parte di Wicor Nixdorf un requisito implicito poteva essere la necessità di uno strumento o applicativo che non necessitasse più di tanto un continuo intervento da parte dei suoi dipendenti.

2.1.2 Modalità di realizzazione

Realizzazione o acquisizione (con esame e valutazione delle eventuali alternative)

Come già spiegato nella sezione 1.1.2.2, Wincor Nixdorf non ha un approccio simile a quello delle software house, quindi di sicuro la scelta strategica era di puntare su prodotti già realizzati, e customizzarli per il cliente, piuttosto che svilupparne altri ex-novo.

Per Primafrost in particolare era stata già realizzata un'applicazione per la gestione dei carichi dei mezzi e per la visualizzazione delle gite (tra l'altro già integrata con Me.R.Sy.). Sembrava quindi logico cercare di integrare il tutto con un'applicazione/servizio che calcolasse in automatico le gite in base a i carichi. La visualizzazione delle gite veniva fatta con una versione di "PTV Maps & Guide".

Era quindi logico cercare una soluzione applicativa sempre progettata da PTV, sia perchè poteva risultare più semplice da integrare sia perchè PTV al momento è una delle aziende leader nel campo.

2.1.3 Valutazione delle alternative

Le 3 alternative possibili erano le seguenti:

1. ottimizzare le gite usando gli XServer
2. ottimizzare le gite usando PTV Intertour
3. lasciare tutto com'era (non era una vera alternativa, ma era piuttosto l'ultima spiaggia)

L'alternativa era dunque tecnologica, in quanto bisognava solo decidere se l'ottimizzazione delle gite doveva essere fatta automaticamente piuttosto che manualmente, e se era meglio utilizzare Web Service o un approccio applicativo classico.

2.2 Il progetto proposto

Come visto nella sezione precedente, molti aspetti partivano già inquadrati, e tuttavia non era chiaro quale fosse l'alternativa vincente.

Ci si è quindi informati prima sui processi del cliente coinvolti, e si è cercato di sconvolgerli il meno possibile, come spiegato nella prima sottosezione.

Per alcuni problemi secondari, di cui parliamo a breve, sembrava che la componente umana fosse imprescindibile in alcuni casi. Inoltre non era chiaro cosa un'architettura orientata ai servizi aggiungesse all'architettura, per cui si era partiti facendo test utilizzando l'applicazione stand-alone. Chiaro non era neppure quanto tempo ci volesse ad eseguire una corretta configurazione dello strumento, e se questa configurazione potesse essere unica nonostante la variabilità degli scenari.

Nel mio lavoro non ero solo, anzi la mia mansione era di affiancare un giovane impiegato (Primo Tilli) che era statao incaricato di occuparsi del problema. Tutto il lavoro svolto da me e da Primo si trova nelle sottosezioni successive.

2.2.1 Processi del cliente

Sicuramente si può pensare che esista una documentazione ufficiale in cui sono descritti i processi di Primafrost, anche se in modo non necessariamente formale, tuttavia comprensibilmente quella documentazione non ci è stata fornita.

La mia conoscenza dei processi di Primafrost deriva esclusivamente dal bagaglio di conoscenza posseduto da Wincor Nixdorf, e da quanto è emerso negli incontri con il cliente, quindi la visione che ne deriva potrebbe essere certamente limitata, ma a mio parere è abbastanza completa per permettere uno sguardo generale alla metodologia di lavoro.

Il proceso che era interessante per noi era il processo di gestione delle consegne,e questo poteva essere scomposto come segue:

1. Sotto-processo di raccolta dati dei clienti
    - ricezione degli ordini
    - eventuale modifica delle finestre di consegna

2. sotto-processo di preparazione della flotta

   (a) creazione delle gite

   (b) eventuale correzione dei carichi dei veicoli

   (c) assegnazione degli autisti

   (d) carico veicoli

3. Sotto-processo di dispiegamento delle consegne

   - tracciamento dei mezzi in marcia

Come si può notare alcuni di questi sotto-processi ho deciso di suddividerli usando elenchi puntati invece che numerati. Questo deriva dal fatto che non so con certezza in che ordine vengano svolte le attività.

Per quanto conosco infatti Primafrost stabilisce con i punti di consegna delle finestre di servizio, entro le quali la merce deve essere consegnata perchè è presente il personale addetto allo scarico. Consegnare al di fuori di queste finestre risulta quindi in uno spreco di tempo e denaro sia per l'azienda che per il punto di consegna, quindi risultano essere un vincolo operativo.

Periodicamente l'azienda quindi raccoglie gli ordini dai suoi clienti/punti di consegna e da questi dati parte nella preparazione della flotta.

Le gite vengono create su base giornaliera, e chiaramente non è necessario che ogni cliente abbia ordini nuovi ogni giorno, quindi i possibili scenari che ne derivano sono abbastanza dinamici.

Come già accennato in altre sezioni, il processo di creazione delle gite viene svolto prevalentemente a mano da un dipendente (un ex-camionista, per la precisione) che ha grande esperienza delle strade italiane, e quindi può fare previsioni sulle situazioni contingenti che si potrebbero presentare agli autisti in determinate vie ed in determinati momenti della giornata.

Una volta trovato il percorso, è necessario calcolare una stima dei kilometri da percorrere e tempo potenziale impiego di tempo. Almeno questa parte non era eseguita in maniera completamente manuale, ma mediante il software 'Maps&Guide' si riusciva ad ottenere stime abbastanza precise.

Fatte le gite e apportate eventuali correzioni, la flotta ed i carichi sui mezzi vengono preparati, ed una volta partiti i mezzi vengono tracciati temporalmente mediante appositi strumenti, in modo da assicurarsi che gli autisti rispettino i tempi di lvoro e di riposo. Alcuni di essi in particolare sono già tracciati anche mediante GPS, quindi si possono ottenere informazioni ance sulla gita realmente percorsa, anche se tuttavia sono una ristretta minoranza.

2.2.2 Problemi secondari

Un vincolo operatico piuttosto ovvio è rappresentato dalla capacità dei mezzi di trasporto. Le eccezioni per quanto riguarda questo limite non possono che essere gestite in maniera manuale, con conseguente rischio di perdita di ottimalità della soluzione.

Con la vecchia suite capitava ogni tanto una incorretta gestione dei carichi dei camion.

La componenete umana in questo caso si rivelava indispensabile: spesso era necessario modificare le gite già calcolate per far tornare i conti, ed un uomo con una discreta esperienza ed abilità poteva individuare in pochissimo tempo le modifiche che servivano allo scopo senza aumentare i mezzi e senza sconvolgere più di tanto la vecchia pianificazione.

Un'altro problema che persino con i mezzi di oggi non viene trattato nelle applicazioni commerciali è l'eccesso di traffico che alcune strade riscontrano periodicamente durante la giornata, il quale può creare diversi disagi agli autisti.

Ancora una volta un essere umano con un po' di esperienza può riuscire invece a prevedere il problema, e addirittura sa indicare se il problema affligge uno solo dei versi di marcia, e può quindi avvalersene nel tentativo di ottimizzare le gite.

Per quest'ultimo problema PTV sta compiendo degli studi e dei rilevamenti statistici, ma è chiaro che ancora per un po' di tempo l'essere umano è destinato a rimanere indispensabile.

Come se non bastasse le tipologie di merce caricata incidono sul tipo di veicolo da utilizzare, e da quest'ultimo dipende quanta merce di un certo tipo può essere caricata. Ad esempio non è possibile caricare assieme articoli di ortofrutta e salumi, poichè non sarebbe a norma rispetto alle norme vigenti. La merce ortofrutticola infatti ha una certa carica batterica e micotica, che rischia di contaminare il resto della merce, e si rende necessario l'utilizzo di una paratia per dividere i tipi di merce. La paratia è possibile montarla se il mezzo possiede almeno 2 ingressi al vano merci, e quindi questo limita la scelta all'interno della flotta.

Questo genere di problemi di carattere alquanto complesso è ancora risolvibile in maniera semi-automatica, ovvero l'operatore è in grado di indicare una serie di vincoli che chiameremo "qualifiche" sulle merci, i veicoli e persino gli autisti, nel caso si necessitasse di abiliazoni particolari, e lasciare che un software di un qualche tipo risolva il problema delle corrispondenze, ma anche in questo caso la costruzione dei vincoli non può che essere fatta da una persona che conosce la situazione contingente al problema.

2.2.3 Training su PTV Intertour

Per prima cosa si è voluto provare ad usare PTV Intertour, in modo da non complicare da subito la possibile architettura finale.

Per utilizzare l'applicazione è necessario di un'addestramento veramente minimo (fornitoci dai dipendenti di TPS), gran parte dei comandi dell'interfaccia sono abbastanza intuitivi. Giusto per avere una panoramica diciamo che Intertour accetta in input file dati con formato ASCII, XML, CSV oppure anche da tabelle di frontiera in formato SQL SERVER o ORACLE [15].

I dati necessari all'applicazione riguardano depositi (in cui sono inclusi anche gli ordini), veicoli ed autisti. Questi dati vengono chiamati "master data", e vengono memorizzati in locale su filesystem, e successivamente processate dall'applicazione, la quale ricava anche i dati relativi alla geocodifica degli indirizzi.

Da qui PTV Intertour ricava uella che si chiama "matrice delle distanze", ovvero ad ogni coppia di destinazioni associa la distanza del più breve percorso che passa tra le due. questa matrice è utilizzata dall'euristica per trovare la corretta soluzione.

Nei *master data* è possibile indicare quelli che vengono chiamati dall'applicazione 'giri base', ovvero gite precalcolate dall'utente esterno. PTV Intertour permette infatti sia di calcolare le gite ex-novo, sia di una funzionalità per cercare di utilizzare i 'giri base' come punto di partenza nella ricerca di una soluzione ottima.

L'utlima funzionalità che ci hanno illustrato era l'utilizzo di quella che in Ricerca Operativa viene chiamata 'post-ottimizzazione', ovvero un possibile raffinamento ulteriore della soluzione trovata, basata sul componenete MEFISTO (vedi 1.2.13).

Per un'interfacciamento più user-friendly o per semplice integrazione con l'ERP aziendale è possibile inoltre utilizzare un altro applicativo, IT Manager, che permette di caricare facilmente i dati e memorizza tutto in un database in formato SQL SERVER.

L'interfaccia di IT Manager è poco più che minimale, ma contiene tutte le funzionalità necessarie: è possibile modificare i dati, mandare a Intertour solo un sottoinsieme ben preciso,ecc. Inoltre è permette di verificare le coordinate di georeferenziazione disegnando il punto su una mappa.

2.2.4 Configurazione di PTV Intertour

La spiegazione che ci era sata data su questo prodotto era davvero dettagliata, ma tuttavia tutti i dettagli che ci erano stati forniti erano utili solo come aiuto durante lo svolgimento dei casi d'uso più o meno tipici, mentre il funzionamento interno del programma, come ci si poteva aspettare, rimaneva per lo più oscuro.

In questa fase l'attività principale è stata quella di eseguire una attenta esplorazione dell'interfaccia alla ricerca di parametri che ci permettessero di migliorare i risultati. A questa era affiancata una attività di validazione misurando come si comportava in alcuni casi reali.

2.2.4.1 Parametri dell'euristica

Ben presto ci siamo resi conto che il controllo sui risultati poteva essere anche molto fine, anche se abbiamo riscontrato qualche potenziale pecca. Oltre a diversi tool per la visualizzazione dei risultati infatti, l'applicazione ci forniva funzionalità (figura 2.1):

- per cambiare alcune proprietà relative ai mezzi (velocità media, qualifiche, costi per fascia oraria/kilometrica, ecc.)
- per aggiungere/modificare alcuni vincoli per quanto riguardava la soluzione (tempo massimo che un mezzo poteva attendere presso un punto di consegna in attesa che aprisse, costo relativo alla singola unità di tempo, finestre di consegna, ecc.)
- per decidere se adottare i "giri base" come punto di riferimento o meno (era possibile ad esempio decidere di tenere fissi i clienti che dovevano essere visitati nelle gite, ma fare in modo che fosse l'applicativo a decidere qual'era l'ordine che minimizzava i costi)
- per tenere traccia dei vincoli che erano stati eventualmente violati dalla soluzione, mediante una piccola console.

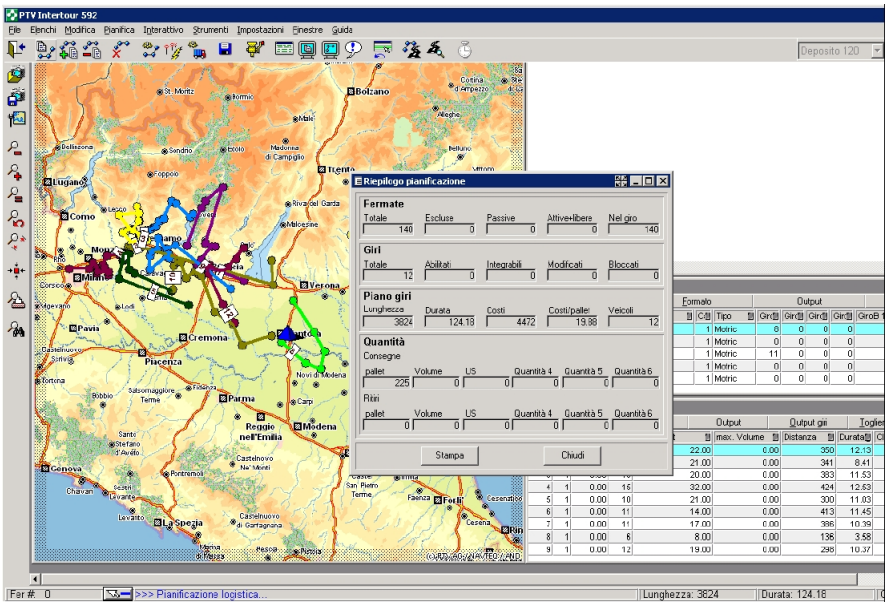


Figura 2.1: Schermata di PTV Intertour

Ognuna di queste funzionalità ci aiutava nella creazione di uno scenario simulato che cercavamo di rendere il più possibile vicino alla realtà. Più lo scenario che andavamo a creare si avvicinava allo scenario del nostro problema, più i risultati che ottenevamo da Intertour sembravano affidabili, o quantomeno convincenti.

Un dettaglio che si rivelato importante successivamente stava nel fatto che i mezzi non erano visti come singole unità, ma come classi di veicoli. Gli attributi relativi al mezzo non appartenenvano al singolo mezzo, ma all'intera classe. Questa particolarità ci è stato spiegato che era stata introdotta per motivi prestazionali: più la flotta era omogenea, più i calcoli si semplificavano, ed una simile rappresentazione forzava a distinguere solo ciò che era chiaramente differente.

2.2.4.2 Qualifiche

Per ogni mezzo era possibile definire arbitrariamente dei flag che rappresentassero le "qualifiche del mezzo", che ci permettevano di risolvere problematiche complesse come quella dell'ortofrutta vista in 2.2.3.

Il meccanismo in sè era semplice, ed il sistema, dopo la prima volta era stato impostato, era in grado di memorizzare le qualifiche attivate in modo da non perderle tra una sessione di utilizzo e l'altra.

Era necessario prima di tutto attivare i flag per i mezzi che si sapeva possedessero certe caratteristiche e poi, ogni volta che un'ordine aveva certe caratteristiche, si attivava il flag corrispondente anche per il cliente (importante notare che la qualifica veniva vista come un'attributo del cliente, e non dei suoi ordini). Dopodichè era l'applicazione a tenere conto delle qualifiche e cercare di attribuire il mezzo giusto durante il calcolo della soluzione. Se il vincolo della qualifica non veniva rispettato dalla soluzione trovata, nella piccola console appariva quale gita aveva

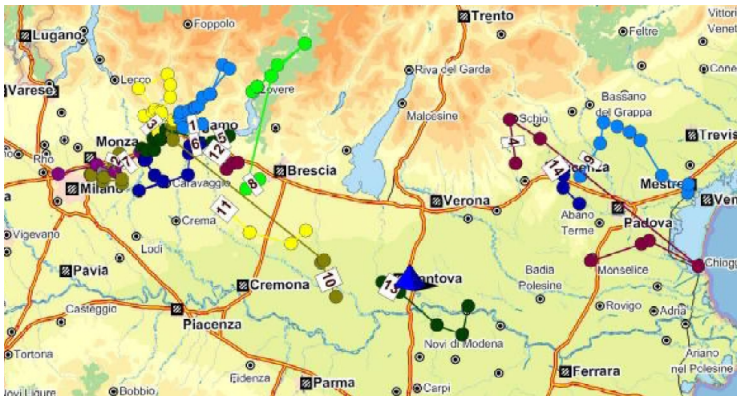


Figura 2.2: Alcune gite mostrate da Intertour

violato il vicolo e su quale punto di consegna. Era anche possibile automatizzare in parte anche le attivazioni delle qualifiche definendo alcune macro, ma nel nostro caso ne abbiamo fatto a meno perchè non ne facevamo un uso così esteso.

2.2.4.3 Visualizzazione dei risultati

I punti di consegna appaiono rappresentati come dei pallini nella mappa che mostra l'applicazione, collegati da linne continue che sono le gite. Le gite si dipartono, come è logico, dal maggazzino di Mantova, che appare come un triangolino. Come triangolini appaiono anche eventuali *transit point*, punti particolari che si comportano sia come normali destinazioni (un certo quantitativo di merce deve esservi consegnato) sia come magazzini (la merce, una volta arrivata, viene smistata utilizzando altri corrieri).

Questa rappresentazione, benchè semplice ed abbastanza intuitiva, nasconde in sè quello che potrebbe essere considerato un errore di usabilità: le linee che congiugono i vari punti di consegna non ricalcano minimamente quello che è il percorso reale, sono semplicemente linee che congiungono coppie di punti nel modo più diretto possibile (figura).

Questo significa, come è succeso in caso reale, che se due punti si trovano ai due estremi di un lago e fanno parte della stessa gita vedremo un linea congiungerli e passare **sopra il lago**. Ad una prima vista questa rappresentazione potrebbe sembrare fuorviante oppure anche segnalare un errore nelle impostazioni, ed è quello che è capitato. Invece, tenendo conto di quanto detto prima, è chiaro che non c'è niente di strano.

Questa rappresentazione potrebbe derivare anche da limiti pratici, in quanto non sappiamo se nella matrice delle distanze vengono memorizzate solo le distanze tra le destinazioni o anche i percorsi che le collegano. In ogni caso non è un difetto fatale, anzi per fortuna la mappa risulta molto utile anche in presenza di questo difetto.

2.2.4.4 Velocità dei mezzi

I dipendenti di TPS ci hanno subito avvertito di una peculiarità di questo prodotto: le zone a ZTL non sono correttamente gestite. Questo significa che

se si fosse presentato ad esempio il caso di un punto di consegna in centro a Milano lo l'applicazione ci avrebbe segnalato che il punto era raggiungibile, ma non avrebbe segnalato che era necessario nell'ultimo tratto utilizzare un mezzo leggero o con autorizzazione. Bisognava quindi stare un attimo attenti ai risultati che lo strumento mostrava alla fine del calcolo.

A mio parere questo difetto sembra come al solito legato al fatto che nella matrice delle distanze non vengano salvati altri dati sul percorso più breve.

Per ovviare al problema, e cercare di far quadrare il più possibile i conti soprattutto per i tempi di viaggio, si doveva intervenire a mano, con 3 approcci possibili:

1. **osservare la gita, calcolare circa il punto in cui il mezzo entrava nella zona ZTL, mettere in quel punto un transit point, e impostare delle qualifiche sul vero punto di consegna in modo che fosse selezionato un mezzo leggero o aurorizzato.** Questo era decisamente l'approccio più corretto, ma molto difficile e dispendioso in termini di tempo. Inoltre ricordiamo che era difficile stimare il punto preciso poichè la mappa non mostrava il vero percorso del mezzo, ma soltanto una linea che congiungeva 2 punti.

2. **stimare più o meno il punto dove il mezzo entrava nella zona ZTL, e stimare la velocità media del mezzo leggero, e modificare la velocità del mezzo pesante in modo che avesse un valore che era una specie di media pesata tra i due. Questo approccio risultava meno corretto, ma sembrava più veloce.** In realtà si perdeva comunque un po' di tempo, poichè i mezzi erano suddivisi in classi, e la velocità media era un'attributo della classe, non del veicolo, per cui sarebbe stato necessario isolare il mezzo scelto e metterlo in una classe a parte.

3. **era possibile modificare gli attributi relativi alla strada in modo da impostare un determinata velocità massima (una specie di limite di velocità).** Questo metodo, molto simile al secondo, in realtà risulterebbe molto più semplice perchè tutti i relativi calcoli verrebbero campiuti interamente dallo strumento, e di sicuro anche relativamente veloce. Tuttavia ancora una volta usando solo la mappa non si poteva capire da che via specifica il mezzo sarebbe passato, quindi era chiaro che si poteva utilizzare questa tecnica solo se ad esempio utilizzavamo Maps&Guide per visualizzare il percorso.

La nostra fortuna era che situazioni come quella sopra descritta non si presentavano, comunque abbiamo deciso di prendere nota del problema. Il vero difetto dal mio punto di vista non consisteva nel fatto che PTV Intertour non era in grado di risolvere agilmente il problema, ma piuttosto il fatto che non potesse nemmeno segnalare all'operatore che era necessario controllare i risultati.

2.2.4.5 Validazione dei risultati

Come test per la configurazione adottata abbiamo utilizzato dei dati provenienti da Primafrost e rilevati in giorno specifico, in modo da fare una prova con uno scenario reale, con tanto di "giri base" (ovvero le gite originali tracciate manualmente).

Sono state fatte più prove, provando a far variare il numero di punti di consegna e la loro disposizione, e controllando ogni volta se i tempi di riposo per i camionisti rispettavano le norme vigenti.

Ciò che si è potuto osservare è che, utilizzando le gite originali come punto di partenza si otteneva comunque un risparmio, misurabile attorno all'1-2%, che era nulla in confronto al caso in cui le gite venivano ricalcolate da capo. in quest'ultimo caso si poteva anche raggiungere il 5% lasciando agire per un po' la post-ottimizzazione.

Come già detto precedentemente in 2.1, queste percentuali possono apparire irrisorie, ma non lo sono affatto, poichè i fattori di costo per questo tipo di attività si aggirando complessivamente sull'ordine dei milioni di euro, e quindi si sta parlando di una discreta somma comunque.

Ciò che abbiamo inoltre rilevato sono le prestazioni del componente applicativo: in al massimo una decina di secondi un problema con circa 200 punti di consegna ed una ventina di mezzi era stato completamente risolto, senza violazioni di vincoli e migliorando la soluzione trovata manualmente del 3% circa (tra i vincoli contavano anche i tempi di riposo).

La post-ottimizzazione impiegava ben di più: in teoria la si poteva lasciare andare quanto si voleva, ma sperimentalmente si era notato che i miglioramenti più notevoli si avevano già entro i primi 2 minuti. Passato questo limite, il tempo tra un miglioramento ed il successivo tendeva ad allungarsi troppo, per cui si è deciso di impostare come condizione di terminazione il fatto che fossero passati 120 secondi. Pur usando un tempo così ristretto, si notava che in media la post-ottimizzazione migliorava la soluzione trovata dall'euristica dell'1-2% al massimo.

2.2.5 Comportamento dello strumento in caso di scenari dinamici

Prima di comunicare ufficialmente al cliente i dati che avevamo raccolto, era necessario fare anche delle prove con dati che provenissero da giornate diverse, in modo da vedere come lo strumento si comportava di fronte a scenari dinamici.

Per nostra fortuna il cliente era molto ben disposto nei nostri confronti, ed erano già disponibili i dati di alcune giornate, già riversati all'interno dell'ERP aziendale.

2.2.5.1 Problemi di formato sui dati in input

In questa fase abbiamo avuto qualche problema con il formato dei dati in input per Intertour. Me.R.Sy. infatti era in grado di restituirci tutti i dati che ci servivano in formato CSV, ma IT Manager, che noi usavamo per questa procedura, voleva che le colonne fossero in un certo ordine, mentre Intertour voleva che le coordinate per la georeferenziazione fossero nel formato Mercatore.

In sé e per sé questo problema era assai stupido, e forse si poteva risolvere anche solo con qualche macro di Microsoft Excel. Questa infatti è stata la prima strada che abbiamo seguito, ma ben presto ci siamo resi conto che i dati che provenivan dall'ERP non riguardavano solo le giornate che servivano a noi. Era dunque necessario filtrare i record ottenuti, e poi applicare tutte le trasformazioni necessarie. Con le macro di Excel era possibile eseguire le varie operazioni singolarmente, ma visto che avevamo più tabelle (quella dei

mezzi e quella degli ordini dei clienti), la cosa si faceva più difficile, perchè da una parte dovevamo riuscire ad escludere più dati possibile, mentre dall'altra non potevamo permetterci di far saltare i controlli dei riferimenti incrociati, per cui dovevamo eliminare solo lo stretto necessario. Per farla breve avevamo veramente bisogno di un'operatore che funzionasse più o meno come il JOIN di SQL per eseguire tutta la procedura in modo da non fare danni.

Così abbiamo provato ad usare le tabelle Excel come input per altri programmi che ci permettessero di utilizzare qualche operazione similare ai DBMS. Dato che l'ambiente in cui ci trovavamo a lavorare aveva un spiccata dominanza di sistemi e applicazioni Windows-like, abbiamo fatto una prova con MS Access, visto che apparteneva alla stessa suite.

Ben presto però ci siamo resi conto dei limiti che l'applicazione aveva, in quanto permetteva di importare le tabelle Excel, ma non permetteva di usare operatore come il JOIN perchè di fatto non stava utilizzando veri database.

Così l'alternativa è stata l'utilizzo di SQL Server 2005 Express Edition, per via del fatto che comunque è un DBMS, è free, ed è della stessa casa, per cui potevamo aspettarci un po' di compatibilità.

Indagando un po' abbiamo infatti trovato una pratica funzionalità che ci permetteva di importare direttamente i dati dei file Excel all'interno delle tabelle del database e viceversa.

In questo modo siamo riusciti a risolvere tutto il problema in appena mezza giornata.

2.2.5.2 Nuove gite, calcolate su più giorni

Con i nuovi dati abbiamo potuto fare qualche test anche su pianificazioni di altre giornata di consegna. I risultati per fortuna si mantenevano nella media di quanto precedetemente osservato.

Con queste ultime prove avevamo dimostrato che lo strumento si comportava bene anche in presenza di scenari dinamici, senza cambiare impostazioni.

2.2.6 Prima presentazione dei risultati al cliente

Abbiamo potuto finalmente presentare in maniera ufficiale i risultati ottenuti in un meeting con tutti gli stakeholder riuniti. Oltre ad una rappresentanza di Wincor Nixdorf e di Primafrost erano presenti anche TPS e gruppo Lombardini. Quest'ultimo in particolare era presente in quanto voleva conoscere quanto il metodo di consegna di Primafrost sarebbe cambiato, visto che l'azienda rifornisce i suoi punti di consegna.

L'incontro è proceduto abbastanza bene, il cliente ha confermato che dal suo punto di vista si stava andando nella direzione giusta.

Alcune gite di quelle originali tuttavia sembravano strane perchè erano stati registrati dei tempi che lo strumento non poteva riprodurre, così gli esperti di TPS e i dipendenti di Primafrost e Wincor Nixdorf si sono ripromessi di ricontrollare i dati per quelle gite in particolare.

E' emerso anche un'altro dato interessante, di cui però e meglio parlare nella sezione 2.3.

2.2.7 Percorso più corto o più breve?

Una sola domanda rimase in sospeso dopo questo primo incontro. I dirigenti di Primafrost, in linea del tutto teorica, si chiedevano se era possibile regolare lo strumento in modo che invece di utlizzare i percorsi più brevi tra i punti di consegna, utilizzasse invece quelli più veloci.

Questa richiesta era dovuta principalmente alle modalità di pagamento degli autisti: alcuni erano pagati al kilometro, altri all'ora, quindi non era così irragionevole cercare di minimizzare le distanze nel primo caso ed i tempi nel secondo.

Per riuscire a soddisfare la richiesta dovevamo sapere di più sull'architettura interna del programma e capire come manipolare i parametri dell'euristica in modo che tirasse fuori i risultati che ci servivano.

2.2.7.1 Il componente MEFISTO

MEFISTO è stato l'unico componente che ci appariva manipolabile da questo punto di vista, ed è stato anche l'unico componente interno di cui abbiamo ricevuto un minimo di documentazione da parte di TPS.

Il funzionamento in dettaglio di MEFISTO può essere trovato nella sezione 1.2.13. Il suo funzionamento può essere riassunto in poche parole: questo componente parte dal una soluzione base trovata da un'euristica veloce e cerca di migliorarla. I pesi servono ad indirizzare la ricerca e quindi a determinare l'aspetto della soluzione finale.

I pesi non erano modificabili da interfaccia grafica, ed ecco perchè non li avevamo notati finora, ma da un file di configurazione, la cui locazione ci è stata indicata dai tecnici TPS.

Ciò che si è potuto notare è che, ponendo i risultati in un ipotetico piano cartesiano organizzato come in figura 2.3 e 2.4, la funzione risulta abbastanza prevedibile, tuttavia in alcuni casi che abbiamo trovato presentava alcune discontinuità.

Non era pertanto possibile prevedere in modo accurato i risultati che si sarebbero ottenuti.

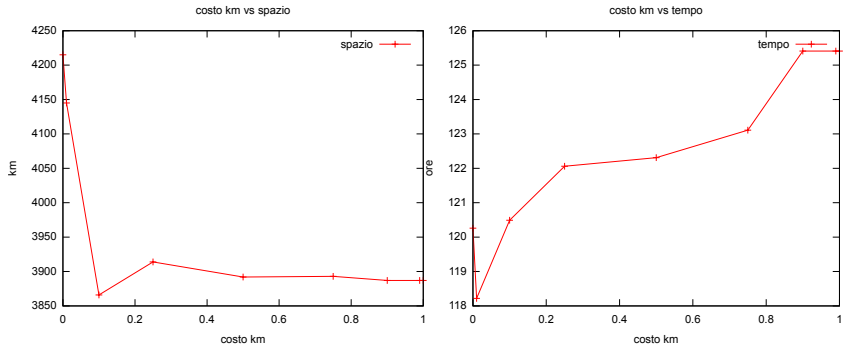
2.2.7.2 Le gite che non tornavano

Oltre al problema della non completa prevedibilità dell'influenza dei fattori di costo, c'erano anche da tenere in considerazione le gite il cui costo non tornava. sembrava infatti che i ottenuti dal precedente sistema, che usava Maps&Guide, non collimassero con i dati ottenuti utilizzando PTV Intertour. In particolare quelli ottenuti con Maps&guide sembravano in alcuni casi migliori.

La prima ipotesi da verificare era che stessero usando la stessa versione delle mappe, altrimenti era chiaro che da dati diversi derivassero risultati diversi.

Sono stati controllati quindi i percorsi di riferimento alle mappe, ma erano identici, e ce lo hanno confermato anche i tecnici TPS.Siamo pertanto tornati a controllare se esistevano altri parametri che influenzassero il calcolo e che fossero comuni ai due sistemi. A prima vista però questa strada sembrava un vicolo cieco.

La soluzione di questo problema venne data da Cristian, il dipendente di Primafrost che si occupava del calcolo delle gite. Mentre noi cercavamo di riprodurre i suoi risultati, lui aveva cercato di riprodurre i nostri, con successo.



(a) (b)

Figura 2.3: *Rappresentazione grafica dell'andamento dei risultati dell'euristica*
Come si può notare, sia dal punto di vista spaziale che di quello temporale, la funzione risulta in linea di massima prevedibile, ma presenta delle discontinuità. In queste figure il costo del tempo di guida era stato impostato su un valore arbitrario di 100. Il valore riportato per il costo dei kilometri è in realtà il rapporto tra il costo dei kilometri e quello del tempo.

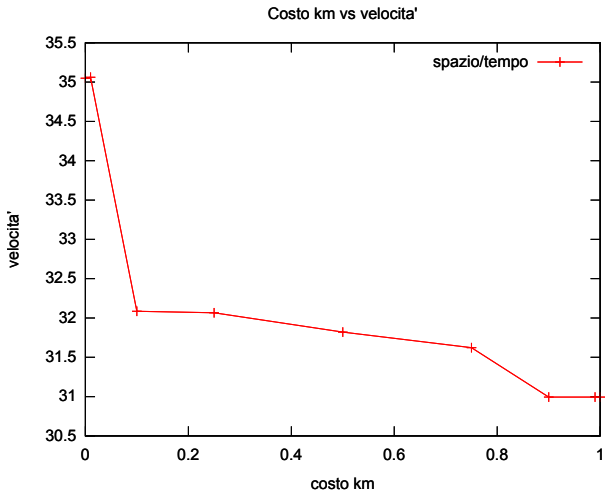


Figura 2.4: *Rappresentazione grafica della relazione tra coefficiente di costo e velocità*
Grafico ricavato da quelli già visti in figura 2.3. Per ogni punto sull'asse delle ascisse, è stato fatto il rapporto tra il valore delle ordinate dei due due grafici, in modo che si ottenesse una misura della velocità media. Come si può notare, con un valore minimo del coeffciente del costo dei chilometri vengano privilegiate soluzioni con una velocità media più alta, mentre mano a mano che il costo dei chilometri cresce, vengono privilegiate soluzioni con velocità medie più basse ma che contemplano percorsi in media più brevi.

Tutto si era risolto con la variazione di un singolo fattore, che dall'interfaccia sembrava effettivamente influenzare il calcolo del percorso, ma solo per quanto riguardava la sua lunghezza.

La figura illustra quello che esponeva l'interfaccia: mettendo un valore minimo a quel coefficiente avremmo in teoria dovuto influenzare la ricerca in modo che selezionasse il percorso più veloce, al contrario avrebbe selezionato il più breve.

Non è dato sapere di preciso come questo singolo fattore influenzasse il calcolo, ma si possono fare supposizioni: supponendo che l'algoritmo utilizzato da entrambi i programmi sia simile a grnular Tabu Search, quel fattore potrebbe in realtà aver influenzato la "granularità" dello spazio delle soluzioni, cambiando il numero di archi presi in considerazione. Una volta trovato quindi una disposizione degli archi simile, i due programmi avrebbero cominciato a restituire soluzioni simili tra loro.

2.2.7.3 La ricerca dei giusti valori per i fattori

Risolto il problema delle gite "non corrette", rimaneva il problema di trovare i valori "migliori" per soddisfare le esigenze del cliente.

Vi erano sostanzialmente 2 metodi possibili:

1. Cercare manualmente, attraverso una serie di prove, i valori corretti

2. Cercare di automatizzare il processo descritto al punto 1

Ognuna delle due scelte aveva vantaggi e svantaggi, ovviamente. Nel caso della ricerca manuale i tempi potevano essere anche molto lunghi e la qualità della soluzione trovata scadente, viceversa per il caso automatico, con la differenza che per automatizzare il processo e mantenere i tempi ragionevolmente brevi potevano essere necessarie risorse computazionali aggiuntive, all'epoca non disponibili.

Si è quindi optato per la prima scelta, raggiungendo una soluzione abbastanza buona dopo qualche giorno di lavoro.

Giusto per completezza, i risultati che si sarebbero potuti ottenere con una procedura automatica, sono spiegati nel capitolo 3, 3.

2.2.7.4 Validazione dei risultati

Tutte prove di cui si è parlato prima sono state effettuate sulle mappe e sugli ordini su ci sono anche state effettuate anche le prove fatte per laprima consegna, per cui eravamo piuttosto sicuri dei risultati.

2.2.8 Presentazione dei risultati definitiva al cliente

Il cliente, già contento perchè aveva intuito che si stava andando nella direzione giusta, è stato molto felice di sapere che c'era anche un modo di manipolare il risultato finale dell'applicazione. In questo modo se un domani fossero cambiate le esigenze, si poteva semplicemente agire sui parametri invece che gettare via tutto.

2.3 Analisi costi-benefici

A conti fatti le due architetture eran più o meno equivalenti dal punto di vista dei vantaggi e degli svantaggi.

Ciò che ha portato quindi ad una scelta netta è stato ciò che è emerso già dal primo incontro con gli stakeholder, ovvero il fatto che alcune informazioni non erano del tutto aggiornate all'interno delle mappe.

2.3.1 Vantaggi e svataggi utilizzzando PTV Intertour

Utilizzando Intertour i vantaggi che se ne ottenevano erano valutabili a livello di maggiore semplicità di utilizzo per l'utente finale se si utilizzava direttamente l'interfaccia grafica prevista dallo strumento, ed un costo minore o quantomeno comparabile a quello degli XServer.

D'altro canto l'applicazione non risultava così facile da integrare nell'architettura già esistente, infatti necessitava di qualche workaround solo per interfacciarlo direttamente con l'ERP.

Questa risultava essere un'ottima alternativa nel caso si fosse visto che la componente umana non poteva essere esclusa, in quanto anche se più difficile da integrare poteva risultare più usabile.

2.3.2 Vantaggi e svantaggi utilizzando PTV XTour

Utilizzando XTour era probabilmente più facile da integrare con la'rchitettura preesistente, ma non avrebbe necessariamente semplificato l'interazione con l'utente.

Grazie all'architettura basata su Web Service, si poteva riuscire a visualizzare le mappe anche all'interno di un browser con poco sforzo, visto che esistevano già librerie apposite.

Il problema sarebbe stato dover disegnare ex-novo una nuova interfaccia per l'utente, con possibili problemi di usabilità e di familiarizzazione con i nuovi controlli.

Inoltre non era detto che il costo fosse per forza contenuto, non sono mai state fatte abbastanza prove per avere la certezza se ogni XServer fosse più o meno indipendente dagli altri o se fosse necessario l'utilizzo di più di uno di questi prodotti in sinergia. Facendo conto che le licenze erano diverse per ogni prodotto, significava potenzialmente pagarle tutte a parte singolarmente.

Questa poteva tuttavia essere un'ottima alternativa nel caso si fosse visto che la componente umana era escludibile quasi del tutto. Riducendo i controlli all'osso ed integrando tutto il resto, la qualità delle soluzioni era garantita.

2.3.3 L'incompletezza delle mappe

Già durante il primo incontro con gli stakeholder, come accennato nel capitolo 2.2.6, emerse subito un problema relativo al calcolo dei percorsi.

PTV Intertour infatti non ci permetteva di vedere di preciso quale percorso seguivano i mezzi, e Christian, che in Primafrost si occupava del calcolo delle gite, ha voluto vederci chiaro.

Ricalcolando i percorsi utilizzando Maps&Guide si ottenevano risultati un po' strani infatti (e queste erano le gite che non tornavano), mentre altri sembravano potenzialmente ignorare alcune restrizioni presenti sul manto stradale: in un caso particolare Christian ci ha spiegato che per percorrere una certa di distanza in così poco tempo l'applicazione poteva potenzialmente aver deciso di passare per una galleria attaverso la quale un camion non poteva passare.

Ci siamo sentiti subito di escludere il caso, anche se sapevamo che Cristian aveva più esperienza di tutti noi in fatto di strade. In realtà parlandone un po' è uscito fuori che l'ipotesi di Cristian non era così insensata, in quanto la stessa TPS ha dovuto confessare che non era così garantito che le mappe fossero accurate al 100%: erano sicuramete complete ed aggiornate per quanto riguardava la cartografia, ma i dati riguardanti la viabilità erano anche frutto di PTV, e poteva darsi che l'Italia non fosse sempre così aggiornata.

2.3.4 Scelta finale

In realtà, per quanto il problema della completezza delle mappe sembrasse un problema minore, aveva rivelato una criticità: non potendo più garantire che i percorsi calcolati avrebbero rispettato di base i vincoli imposti dal codice della strada, non si poteva spingere più di tanto con l'ottimizzazione, anzi bisognava cercare di rifarsi il più possibile ai percorsi precalcolati in origine.

Non si è voluto comunque abbandonare il progetto, in quanto le premesse sembravano promettenti, ma era ormai chiaro che la componente umana non era eliminabile.

Si è quindi deciso di mantenere come applicazione di riferimento PTV Intertour, e di integrarlo in modo da fornire a Cristian tutti i controlli necessari. Con questo assetto la figura professionale di questo dipendente non veniva sostituita, anzi si permetteva all'individuo di crescere ulteriormente nella sua carriera professionale, diventando esperto nell'utilizzo dello strumento.

Questo permetterebbe un domani di poter in teoria affiancare a Cristian un assistente per poter gestire le consegne in caso di sua assenza, ed inoltre con il potenziale miglioramento dello strumento si potrebbero anche raggiungere risparmi sempre migliori, o almeno garantire che più o meno sarà sempre possibile un piccolo margine di risparmio.

Gli XServer non verranno pertanto impiegati, in quanto non offrirebbero i vantaggi sperati e rischierebbero di costare troppo ora le licenze. Questa alternativa sarà rivalutata in futuro, quando magari più clienti utilizzeranno questo servizio e ci sarà bisogno di gestire configurazioni diverse.

Capitolo 3

Implementazione

Nell vicenda spiegata nella sezione 2.2.7.3 rimaneva un dubbio irrisolto: si poteva riuscire a trovare i coefficienti giusti in un modo automatico o comunque semi-automatico?

Questo ci avrebbe permesso di avere un minimo di sicurezza per quanto riguardava l'ottimalità della soluzione trovata e ci avrebbe risparmiato potenzialmente un sacco di fatica.

In questo capitolo presento un mio contributo completamente personale in cui ho cercato di rispondere questo quesito.

Nella prima sezione parlerò in generale delle motivazioni che mi hanno fatto scegliere l'approccio che poi ho usato, quindi descriverò come ho ottenuto i dati, i risultati ottenuti ed infine come ho potuto validare i risultati.

3.1 Approccio generale

Era chiaro che non avrei potuto risolvere il problema utilizzando PV Intertour, in quanto nella documentazione di questo strumento non era minimamente spiegato se vi fosse un modo di impartire comandi all'applicazione senza passare per l'interfaccia grafica, quindi non era possibile utilizzarlo da riga d comando o mediante script di qualche genere.

La selta quindi più ovvia era quella di utilizzare gli XServer, ed in particolare XTour in quanto, sempre stando alla documentazione, era il più simile da un punto di vista di funzionalità ed utilizzo.

TPS aveva fornito a Wincor un account per riuscire a scaricare le ultime versioni degli XServer direttamente dal sito di PTV, per cui questa parte non cosutuiva un problema.

I problemi sono cominciati quando, facendo delle prove con i vari applicativi, mi sono reso conto ingenuamente che le mappe non erano incluse nel download. Chiedendo a TPS mi sono reso conto che le mappe avrebbero occupato circa 10 GB, quindi non era possibile scambiarci i file.

Ho abbandonato quindi quella strada, ma ho ottenuto lo stesso i dati, come spiegherò più avanti, poichè in ogni caso non esisteva un modo per importarli all'interno di XTour.

Ho deciso perciò di eseguire un semplice "dimostrazione", ovvero un test che avrebbe dato risultati da interpretare in linea di massima, visto che ogni soluzione con un più ampio margine di precisione era da scartare.

Il mio approccio è stato quindi quello di reimplementare Granular Tabu Search e di compiere la ricerca dei fattori migliori atraverso un algoritmo genetico, per vedere se quest'ultimo mi aiutava a selezionare i coefficienti più promettenti in un tempo ragionevole.

La ragionevolezza del tempo impiegato l'ho misurata confrontando i risultati ottenuti dall'algoritmo genetico rispetto ad un algoritmo "brute force", che invece tentava tutte le combinazioni dei fattori possibili, scegliendo alla fine il risultato più vicino a quello desiderato. Per motivi di limitazioni di tempo ho deciso di limitare arbitrariamente l'intervallo dei valori possibili per ciascun fattore di costo all'intervallo [0-100]. Questo mi forniva un numero abbastanza grande di coppie possibili da non far perdere troppa generalità al test, senza dover attendere troppo tempo mentre attendevo i risultati.

3.1.1 Reimplementazione di Granular Tabu Search

Per reimplementare il Granular Tabu Search ho utilizzato direttamente l'articolo originale di Toth e Vigo.

Mi sono permesso di fare solo 3 modifiche all'articolo originale, in modo renderlo più simile a quello ce veniva descritto nella documentazione di MEFISTO:

1. Invece di utilizzare solo mosse OR1, ho aggiunto anche mosse OR2 e SWAP, in modo da ampliare la gamma delle possibili variazioni

2. Ho fatto in modo che i valori iniziali dei fattori di costo potessero essere decisi a riga di comando quando si lanciava il programma, in questo modo si poteva utilizzarlo con l'algoritmo genetico

3. Ho deciso di implementare tutto in Python, invece che il C o il C++

Come metaeuristica Granular Tabu Search ha bisogno di partire da una soluzione fornita da un euristica rapida. Nell'articolo viene utilizzata l'euristica di Clarke e Wright, quindi ho deciso di tenere buona quell'implementazione, anche se avrei potuto utilizzare qualunque altra euristica conosciuta.

L'implementazione più prestante si sarebbe potuta ottenere certamente scrivendo tutto in C e ottimizzando per l'architettura su cui l'applicazione avrebbe dovuto girare.

Tuttavia il tempo per sviluppare l'intero codice sarebbe cresciuto troppo rispetto alle mie aspettative: C non possiede nativamente il supporto per liste, mappe di hash e iteratori, e tra l'altro benchè sia un linguaggio molto potente non prevede meccanismi per lavorare con gli oggetti senza allocare/deallocare la memoria manualmente, esponendo quindi il sistema a problemi di memory leak e salti ad aree di memoria non più allocate. Per correggere tutti i bug relativi alla gestione della memoria da soli a mio avviso sarebbe servita una settimana, con il rischio di bug non scovati ed effetti più o meno bizantini anche alla fine della revisione.

Avrei potuto puntare sul C++, ma conoscevo meglio Python, e sapevo che comunque anche il codice Python veniva compilato, anche se magari non era

ottimizzato. Inoltre questo linguaggio aveva tutti i vantaggi che C non aveva, con l'unico difetto che non sapevo prevedere quali performance il codice finale avrebbe avuto. Quest'ultimo problema non era così importante, in fondo la mia era solo una dimostrazione: era importante che fosse corretta, non prestante.

3.1.2 L'algoritmo genetico

Ogni algoritmo genetico (come spiegato precedentmente in 1.4) non fa altro che rappresentare i geni degli individui mediante stringhe, calcolando per ogni stringa un vlore di fitness, ed infine, mediante un processo di selezione, fa accopiare gli individui migliori in modo che la nuova generazione sia migliore di quella prima. Nella riproduzione i corredi cromosomici dei genitori vengono mischiati tramite cross-over, in modo da ampliare lo spettro delle possibili soluzioni, ed infine c'è una probabilità di ottenre una mutazione, che aiuta a non rimanere ancorati ai "minimi locali".

Se come geni pendiamo i coefficienti che usiamo per il granular tabu search, la rappresentazione in stringa di una specifica coppia di valori possibili può rappresentare il corredo genetico di un'individuo.

Nel caso che ci si presentava in azienda noi dovevamo cercare di riprodurre i risultati di Primafrost, trovando i coefficienti opportuni.

Possiamo quindi fare in modo che individui che ottengono i risultati vicini ad un valore preimpostato siano favoriti nel processo di selezione, in modo da ottenere una popolazione finale che presenti come geni solo i coefficienti più promettenti.

Ad ogni ciclo dell'algoritmo da me progettato infatti si ha:

1. inizializzazione degli agenti
2. calcolo della soluzione del problema mediante Granular Tabu Search (i coefficienti usati nella ricerca sono quelli presenti nei geni dell'individuo)
3. selezione degli individui in base ai risultati ottenuti (metodo della ruota della roulette)
4. Cross-over (uniforme) e mutazione

3.2 Esperimenti eseguiti

Per eseguire degli esperimenti i cui risultati potessero essere un minimo validati e confrontati tra loro ho deciso di utilizzare 2 istanze "ideali" del problema, ovvero istanze utilizzate tipicamente in letteratura per confrontare gli algoritmi ma che non hanno pretesa di realtà, ed 1 caso reale, ovvero il caso reale, di cui ho dovuto calcolare la matrice delle distanze, come descritto nella sezione 3.2.1.

Le istanze "ideali" invece le ho potute reperire su *VRP Web* [18], un sito molto completo in cui sono raccolte molte di queste istanze, in modo da promuovere il loro utilizzo negli articoli scientifici che trattano del settore. Sono inoltre suddivise per creatore, e vengono riportati i risultati migliori mai raggiunti da un'euristica istanza per istanza. Visto che il sito è aggiornato con una certa frequenza, i dati che sono raccolti in queste pagine mi sono sembrati affidabili.

Negli esperimenti che avevo in programma ho aggiunto anche un paio di esperimenti "brute force", che non avrebbero operato seguendo l'algoritmo genetico come solito, ma avrebbero invece esplorato iterativamente, coppia per coppia, tutto lo spazio delle soluzioni, e preso la più vicina a quella desiderata.

3.2.1 Ottenere i dati di input

Nel cercare di vedere come l'algoritmo genetico si sarebbe comportato in caso reale dovevo risolvere il problema dell'incompletezza dei dati in mio possesso.

Possedevo infatti i risultati finali di quell'istanza, conoscevo le quantità degli ordini e i mezzi disponibili, ma non conoscevo la posizione geografica dei vari clienti nè tantomeno le distanze reciproche tra di loro.

Avevo provato a farmi restituire tali dati da PTV Intertour, ma non avevo trovato la funzionalità adatta, anche perché, come mi confermarono le persone di TPS poi, quella funzionalità non sembrava essere stata prevista.

La funzionalità di importazione/esportazione della matrce delle distanze era infatti presente solo in un prodotto della serie XServer, XDima. Ho provato quindi, utilizzando delle licenze di prova, a far funzionare tale server, ma infruttuosamente, poiché nel download non erano state previste le mappe geocodificate, e non vi era un modo pratico nè per me nè per le persone di TPS di passarmi tali file, che in totale occupavano approssimativamente una decina di GigaBytes.

Il fatto che non si potessero ottenere i dati direttamente dalle mappe di PTV tuttavia non rappresentava un problema. avevo infatti trovato un simpatico tutorial sugli XServer bastato su Javascript, che permetteva di editare il codice [16].

Studiando un po' l'API pubblica descritta nella documentazione di PTV, ed avendo ancora un file Excel di quelli usati per PTV Intertour, sono riuscito ad impostare il programma in modo che, date una serie di città, calcolasse la lunghezza e la durata del percorso tra ogni coppia di città.

Il risultato di quest'operazione era una sorta di matrice delle distanze, ottima per il gnere di problema che doveva essere trattato.

Visto che i punti d consegna erano dell'ordine del centinaio almeno, l'operazione ha richiesto qualche ora.

A causa di un limite sulla massima granularità nella mappa del tutorial, punti di consegna ppartenenti alla stessa città collessavano. Ho eliminato quindi i punti coincidenti, in quanto erano rendevano solo più pesanti i calcoli da effettuare.

La matrice delle distanze che ho potuto utilizzare dopo quest'operazione risulta quindi un po' diversa da quella reale, ma mi permetteva comunque di fare qualche prova anche su un caso reale.

3.2.2 Risultati ottenuti

3.2.2.1 Esperimenti con istanze CVRPTW

Il mio primo esperimento è stato effettuato sull'istanza 'c101' di Condreau. Questa infatti era una buona istanza di CVRPTW, con 100 clienti ed 1 magazzino.

Sorprendentemente per me, la versione di Granular Tabu Search che avevo implementato non riusciva a risolvere questo problema, o meglio generava solo soluzioni inammissibili, che violavano almeno un vincolo. La prima spiegazione che mi diedi fu che affettivamente il vincolo temporale di 2 minuti fosse troppo stringente, ed il runtime python troppo lento.

3.2.2.2 Esperimenti con istanze CVRP

Dati i risultati ottenuti in precedenza, ho deciso di l'istanza 'c101' di Condreau, ma di considerarla come se fosse un'istanza di CVRP, piuttosto che di CVRPTW, ovvero ho fatto in modo che Granular Tabu Search non tenesse più conto delle finestre di consegna.

In questo modo le soluzioni sono tornate ad essere ammissibili, e pure non troppo lontane dal valore ottimale, come viene mostrato nella sezione di validazione 3.2.3.

Ho scelto quindi anche un'istanza diversa, presa anche da un autore diverso, in modo da evitare possibili schemi ricorrenti inseriti che ogni autore potrebbe avere, e ho scelto l'istanza 'RC101' di Solomon, che aveva in comune con quella di Condreau il numero di clienti da servire, sempre considerandola come un'istanza di CVRP. Anche in questo caso i risultati erano ammissibili, e non lontani dalla soluzione migliore.

Lanciando l'algoritmo "brute force" su entrambi i problemi ho scoperto paio di cose interessanti, visibili anche in figura:

1. ponendo di poter stabilire un ordinamento tra le coppie di fattori, in modo da poterli rappresentare sull'asse delle ascisse, sull'asse delle ordinate ci troveremmo di fronte a quella che sembra una funzione costante, tranne qualche eventuale punto di discontinuità, come visibile in figura.

2. Si può notare come tra la soluzione non post-ottimizzata (quella con coefficienti (0,0), nell'origine) sia sempre peggiore rispetto alla versione su cui invece Granular Tabu Search ha potuto operare

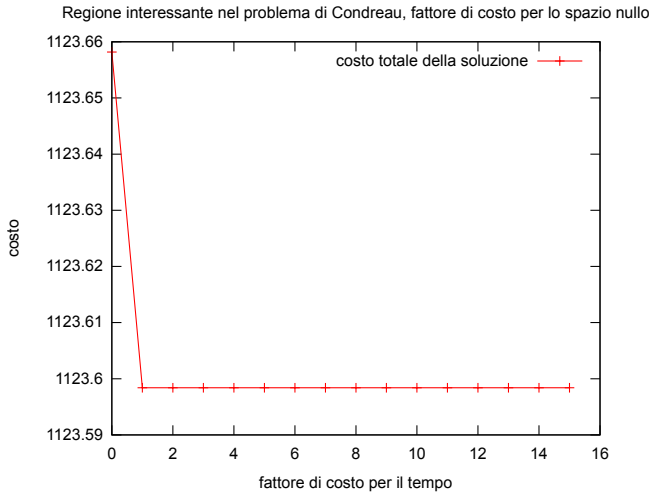
3. Il miglioramento percentuale è tuttavia variabile, e va dallo 0,008% dell'istanza di Condreau al 6% dell'istanza di Solomon, il che più o meno ci riporta in media a quel 3% che era stato osservato nei casi reali

Ho voluto quindi provare a vedere che risultati ottenevo con il caso reale di Primafrost: i risultati sono stati pessimi, in quanto ancora una volta non si riusciva a produrre una soluzione ammissibile.

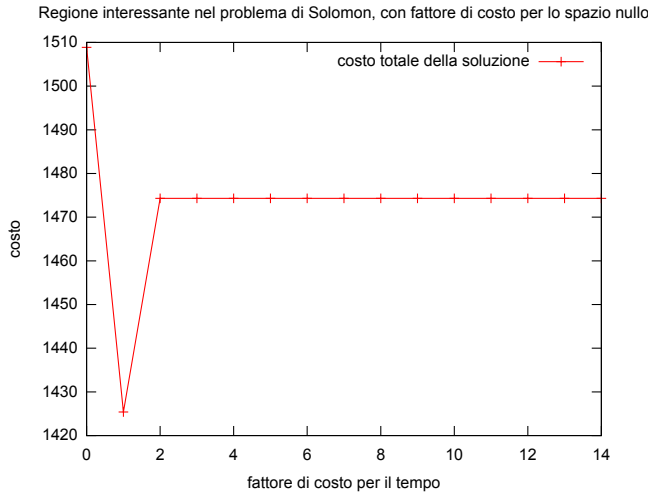
3.2.2.3 La consulenza di un esperto del campo, Paolo Toth

Frustrato dai risultati ottenuti, mi sono rivolto al professor Paolo Toth, presso la Facoltà di Ingegneria di Bologna, in quanto affermato studioso del campo ed inventore dell'algoritmo originale. Dal breve colloquio che ho avuto sono riuscito a capire che non c'era nulla di sbagliato in quello che avevo fatto, ero solo stato troppo ingenuo.

Lo sforzo decisivo in queste applicazioni viene sempre fatto dall'euristica principale, non dalla metaeuristica. L'euristica di Clarke e Wright è veramente molto semplice, per cui può adattarsi bene ai casi della letteratura, ma non così bene a casi reali o semplicemente più complessi. La metaeuristica poi svolge il



(a) Istanza di Condreau



(b) Istanza di Solomon

Figura 3.1: *Comparazione tra due regioni interessanti tra l'istanza di Solomon e quella di Condreau.*
Come si può notare, la rappresentazione dei risultati suggerisce la somiglianza con il grafico prodotto da una funzone costante, meno qualche discontinuità.

suo lavoro tentando di migliorare ancora di più la soluzione ottenuta, ma se la soluzione di patenza non è buona, risulta quasi del tutto inutile.

Utilizzando euristiche più complesse avrei potuto decisamente migliorare i risultati, ma ormai il tempo scarseggiava e non ho potuto fare di meglio.

Questo però ci dice una cosa importante sul prodotto di PTV: con un discreta probabilità l'applicazione da me implementato rispecchia in linea di massima il funzionamento del componente MEFISTO, tuttavia il vero valore dell'intera suite è l'euristica iniziale: se fosse possibile vederne il codice o specifiche di funzionamento varie chiunque potrebbe decidere di competere con PTV in questo settore. Si capisce quindi la motivazione di tanta segretezza sul funzionamento interno dei componenti quindi.

3.2.2.4 Risultati con l'algoritmo genetico

A causa delle scadenze temporali, che ormai si erano rivelate pressanti, ho deciso comunque di fare anche gli esperimenti con l'algoritmo genetico. In fondo dovevo semplicemente dimostrare che questa metodologia poteva comunque produrre i risultati che cercavamo in modo automatico.

Nel nostro caso avevamo dei valori di output da usare come riferimento, e dovevamo trovare quale combinazione di coefficienti produceva quel risultato, o qualcosa di simile, per cui non era così importante quale fosse l'effettivo valore della funzione in ogni punto.

Nel nostro caso i coefficienti su cui potevamo condurre i test erano solo 2: costo dei chilometri percorsi e costo del tempo impiegato. Ogni coppia in teoria poteva produrre soluzioni diverse le une dalle altre, anche se idealmente non era difficile, osservando come si era comportata la funzione sul campo, presagire che maggiore era il valore di un fattore di costo, più la procedura avrebbe cercato di favorire soluzioni che minimizzavano la voce di costo relativa.

Nell'algoritmo genetico venivano utilizzati 100 agenti come "popolazione iniziale" e partire dai 10000 ottenibili facendo il prodotto cartesiano tra i 100 valori possibili per il primo fattore e 100 del secondo.

Ho deciso dunque di limitare il mio studio alle prime 10 generazioni a partire da quella iniziale, in modo da vedere se era possibile già individuare delle tendenze chiare anche se con così pochi agenti e tempo. Al di sopra di questa soglia, infatti, la convenienza del procedimento genetico rispetto a quello "brute force" andava scemando, se si teneva conto anche del fatto che il primo restituiva solo una soluzione approssimata, mentre il secondo una soluzione esatta.

Per rendere evidente che la selezione genetica operava in modo totalmente indipendente dall'andamento della funzione, ho deciso di operare nel seguente modo:

1. Nell'istanza trattata da Condreau, ovrei impostato il valore desiderato in modo che fosse **maggiore** di tutti i valori ottenuti tramite "brute force". In questo modo sarebbe dovuto emergere come più adatto il valore massimo, ovvero quello ottenuto dalla coppia (0,0)

2. Nel caso invece dell'istanza di Solomon, avrei impostato il valore desiderato in modo che fosse **minore** di tutti i valori ottenuti tramite "brute force". In questo modo si sarebbe dovuto notare che la coppia che produceva il valor minimo risultasse avvantaggiata, mentre la coppia (0,0) avrebbe dovuto scomparire, in quanto produceva il valore massimo

Istanza	Costo	Miglior risultato	Post-ottimizato	Miglioramento
Condreau	1123.66	828.54	1123.58	0,001 %
Solomon	1508.89	827.3	1425.4	5,53%
Primafrost	9769558	3753	9769558	0%

Tabella 3.2: Comparazione tra i risultati ottenuti con la mia implementazione e quelli migliori mai trovati

Come si può notare, i risultati ottenuti non sono per niente straordinari, a causa dell'euristica di base troppo semplice, come spiegato nella sezione 3.2.2.3. Tuttavia, tolta l'istanza di Primafrost, che risultava essere troppo complessa e quindi non risolvibile in modo ammissibile in un tempo così limitato, negli altri due casi la metaeuristica produceva comunque un miglioramento, ed in media si riotteneva un 3% di miglioramento rispetto alla soluzione non post-ottimizzata, ovvero un risultato simile a quello che erano stati i risultati con MEFISTO.

3. Non avrei invece minimamente trattato il caso reale in questa ultima fase, in quanto non risultava essere particolarmente significativo

In quest'ultima fase i risultati sono stati abbastanza soddisfacenti, come si può notare in figura 3.3 e 3.2. Per una trattazione dettagliata rimando invece il lettore alla sezione 3.2.3.

3.2.3 Validazione risultati

Ogni esperimento effettuato aveva un qualche criterio di validazione dei risultati.

Per quanto riguarda i primi esperimenti trovati su *VRP Web*, erano già disponibili sul sito i migliori risultati mai ottenuti sulle istanze, e quindi è stata valutata la vicinanza tra i risultati riportati sul sito e quelli ottenuti empiricamente. Per quanto riguarda l'istanza di Primafrost è stata invece valutata la vicinanza con i risultati ottenuti sul campo con MEFISTO. I risultati sono visibili nella tabella 3.2.

Per quanto riguarda i risultati ottenuti con gli algortimi genetici sono state fatte solo delle indagini di tendenza, mediante statistiche, che tuttavia dimostrano come l'algoritmo si comporti piuttosto bene anche se il tempo a disposizione risulta limitato.

La popolazione iniziale era fondamentalmente scelta a caso, ma si notavano un paio di cose interessanti:

1. Se la popolazione inziale non conteneva agenti con corredo genetico (0,0) e (0,1), allora quello che si otteneva era un insieme di agenti che producevano lo stesso valore (come spiegato anche nella sezione precedente), e quindi, essendo tutti equivalenti, nessuno di loro poteva prevalere. Il fatto che il grafico non mostrasse nessuna predominanza di un gruppo su un altro ci diceva che tutti gli agenti, e quindi le coppie di fattori, erano equivalenti

2. Se invece la popolazione iniziale conteneva una delle coppie interessanti, il gruppo che raggiungeva un valore più vicno a quello desiderato tendeva a dominare sugli altri, esattamente come previsto, anche se i risultati potevano variare notevolmente, come si può vedere nelle figure 3.3 e 3.2.

3. I risultati apparivano estremamente evidenti se gli agenti venivano raggruppati in base al valore della funzione di "idoneità" (vedi 1.4). Infatti, a causa delle continue ricombinazioni e mutazioni genetiche, non era possibile studiare nel particolare l'evoluzione dei singoli geni. Raggruppandoli invece in base all'idoneità i risultati apparivano più chiari perchè si riusciva a prevedere quale gruppo avrebbe vinto.

La variabilità dei sisultati di cui parla il punto 2 si riferisce ad un caso abbastanza anomalo che si era verificato in una particolare esecuzione di Granular Tabu Search sull'istanza di Condreau: la popolazione iniziale era ben assortita, e poteva essere divisa in 2 gruppi, uno contenente tutti gli agenti con corredo genetico la coppia (0,0) ed uno contenente tutti gli altri.

Ovviamente (0,0), per come è stato spiegato nella sezione precedente, era una coppia favorita e quella particolare popolazione mostrava un'idoneità più alta. Durante l'esecuzione anomala, a causa di una mutazione genetica in uno dei figli, si era formato un altro individuo (0,0). Questi 2 individui avevano cominciato a riprodursi, oltre che con gli altri, anche tra di loro, siccome erano due individui favoriti.

Casi simili capitavano anche nell'istanza di Solomon, tuttavia, in questa particolare esecuzione, una serie particolarmente fortunata di numeri casuali generati nella procedura di riproduzione avevano dato luogo ad una vera e propria esplosione della popolazione favorita. Dopo questa serie di "rapporti incestuosi", la percentuale degli individui appartenenti alla popolazione con corredo genetico (0,0) era cresciuta molto, e questo favorì nuovamente la riproduzione intra-gruppo rispetto a quella extra-gruppo, essendo la popolazione favorita. Alla decima generazione la popolazione favorita dominava rispetto alle altre.

Questo risultato risulta ancora più straordinario se si pensa che le differenze tra i valori di idoneità tra i 2 gruppi erano situate oltre la quarta cifra significativa.

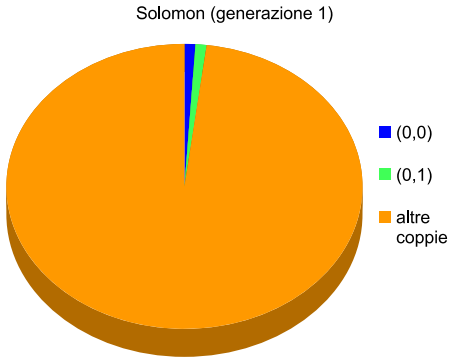
Si poteva quindi concludere che anche la più piccola differenza era importante per l'algoritmo, tuttavia non era possibile stabilire a priori il peso che queste differenze avrebbero avuto sulla singola esecuzione.

3.3 Confronto tra i metodi di scelta dei coefficienti

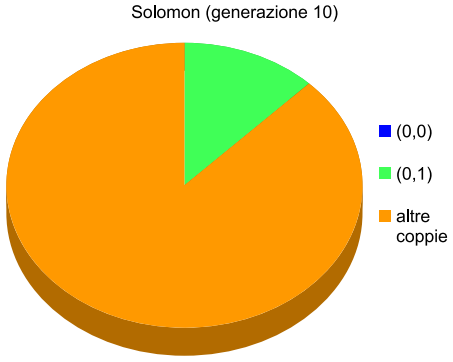
In questo elaborato abbiamo visto 3 metodi con cui si sarebbero potute soddisfare le esigenze del cliente. Ci era infatti stato chiesto di trovare una coppia di coefficienti che permettesse al nostro strumento di riprodurre più o meno fedelmente i risultati già conosciuti.

Sul campo si è potuto vedere all'opera solo uno dei 3 metodi, quello manuale, basato su previsioni e stime fatte da esseri umani. Il metodo si è rivelato vincente poichè ha prodotto buoni risultati in poco tempo, tuttavia non vi è alcune reale garanzia che il risultato trovato sia il migliore ottenibile.

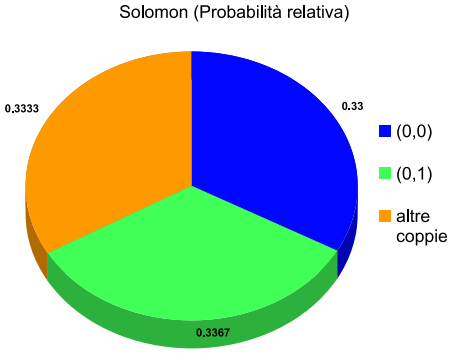
Esistono quindi almeno altri 2 metodi per trovare i coefficienti suddetti, che però fanno affidamento su componenti informatici. Garantiscono un risultato buono, se non necessariamente il migliore in assoluto, però richiedono più risorse per trovare la soluzione giusta (Tabella 3.3).



(a)

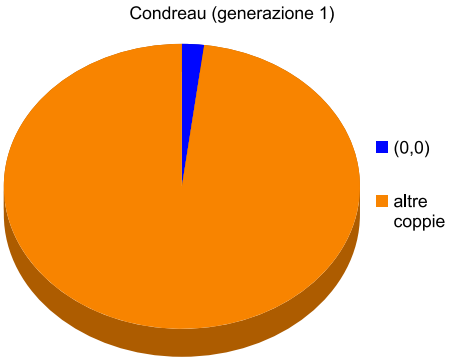


(b)

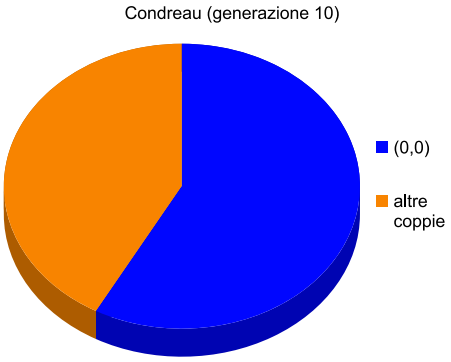


(c)

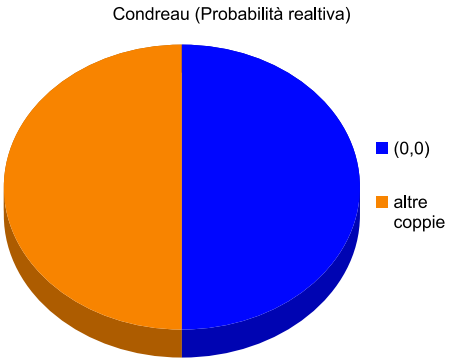
Figura 3.2: *Tendenza della popolazione nell'istanza di Solomon*
Si individuano 3 cluster di "idoneità" (3.2a): quello della coppia (0,0), quello della coppia (0,1) e quelle di tutte le altre coppie. La coppia (0,1) è quella con idoneità maggiore, (0,0) è quella con idoneità minore, mentre tutte le altre presentano un valor medio. Dopo 10 generazioni (3.2b), la popolazione con minor idoneità è già scomparsa, mentre la popolazione più idonea si espande, tuttavia non domina ancora. Questa rapidità nel processo è interessante notando la differenza minima tra le idoneità dei cluster (3.2c).



(a)



(b)



(c)

Figura 3.3: *Tendenza della popolazione in una esecuzione particolare nell'istanza di Condreau.*

Come si può notare in figura, vi sono due gruppi, di cui il gruppo con genoma (0,0) è il favorito (3.3a). A causa di una fortunata serie di eventi, la decima generazione vede lo strapotere di quest'ultimo (3.3b), benché la differenza tra l'idoneità tra i due gruppi fosse differente come valore solo a partire dalla quarta cifra significativa, e quindi nel grafico non si nota (3.3c).

Metodo	Qualità della soluzione	CPU Time (min)
Genetico	Buona (a volte Ottima)	2000
"Brute Force"	Ottima	20000

Tabella 3.3: *Confronto tra le prestazioni dell'algoritmo genetico e quello "brute force".*
L'approccio "brute force" garantisce di trovare una soluzione ottima, ma per trovarla deve esplorare tutte le soluzioni possibili. Nel nostro caso vi erano all'incirca diecimila combinazioni, ognuna delle quali veniva utlizzata all'interno di Granular Tabu Search, la cui esecuzione durava un paio di minuti. Il tempo totale impiegato era quindi di circa 20000 minuti, se eseguito su una singola CPU. Sono state utilizzate circa 40 calcolatori presenti nei laboratori per ridurre il tempo impiegato in modo che risultasse accettabile (circa 3 giorni e mezzo). L'algoritmo genetico impega 10 volte meno tempo, ma non è in grado assicurare che quella restituita sia la soluzione ottima, ma solo che sia una tra la migliori soluzioni esaminate.

I pro e i contro dei vari metodi si possono vedere nella tabella 3.4. Da notare che nel cso dell'algoritmo genetico è possibile distinguere un "caso ottimo" ed un "caso pessimo", questo perché dallo spazio delle 10000 coppie di coefficienti possibili ne sono stati scelti solo 100. Inoltre durante l'esecuzione dell'algoritmo tramite le ricombinazioni e la mutazioni nuove coppie vengono esplorate, per cui possiamo concludere che idealmente, dopo 10 generazioni, o abbiamo ottenuto la dominanza di un gruppo sugli altri, e questo può segnalare il ritrovamento di un'ottima coppia di coefficienti, oppure vengono esplorate un po' meno di 1000 soluzioni equivalenti.

Quindi possiamo dire che, eseguendo più volte l'algoritmo genetico, per esplorare tutte le possibili soluzioni impieghiamo più o meno lo stesso tempo (forse un po' maggiore) rispetto all'algoritmo "brute force".

Una cosa che non emerge dalle tabelle è come cresca il tempo impiegato per trovare una buona soluzione nei 3 metodi. Poniamo ad esempio che aumenti il numero dei coefficienti da controllare e/o gli intervalli di valori possibili per ogni coefficiente.

Nel caso aumentino semplicemente gli intervalli, con il metodo manuale il tempo cresce sicuramente, anche se non sappiamo bene prevedere di quanto. Se invece aumentiamo il numero di coefficienti cambia tutto, in quanto i risultati diventano difficili da rappresentare in forma grafica, e quindi più difficili da comprendere, per cui possiamo pensare che un aumento dei coefficienti sia più critico di un aumento degli intervalli, e potrebbe rivelarsi una crescita non lineare.

Nel caso dell'algoritmo "brute force" non cambia nulla, la crescita di tempo rimane lineare sempre e non si perde niente per quanto riguarda la qualità della soluzione, ma i tempi diventano veramente lunghi.

Nel caso dell'algoritmo genetico invece, la crescita di tempo rimane lineare comunque, usando qualche accorgimento nella scelta della popolazione iniziale, si perde qualcosa dal punto di vista della garanzia dell'ottimalità della soluzione, ma si può riuscire ad impiegare fino a 10 volte di meno, come visto in questi esperimenti, nel caso ottimo.

Metodo	**Qualità soluzione**	**Tempo (min) (Caso Ottimo)**	**Tempo (min) (Caso Pessimo)**
Manuale	Buona	Qualche giorno	Solo stimabile
Genetico	Buona o Ottima	2000	un po' di più di 20000
"Brute Force"	Ottima	20000	20000

Tabella 3.4: *Comparazione tra i 3 approcci.*

Come mostra bene ed in modo sintetico la tabella qui presente, il vantaggio della taratura manuale è la rapidità con cui può produrre un risultato tutto sommato accettabile, ma non garantisce nulla sui tempi impiegati né sulla qualità della soluzione. L'algoritmo "brute force" è molto migliore dal punto di vista della soluzione, ma pessimo come tempo impiegato. L'algoritmo genetico nel caso ottimo e migliore di entrambi, mentre nel caso pessimo presenta le stesse prestazioni dell'algoritmo "brute force".

Conclusioni

Si può quindi concludere che, allo stato odierno, il mondo commerciale è ora in grado di rispondere anche alle esigenze di chi ogni giorno deve gestire un'intera flotta di veicoli e qualche centinaio di punti di consegna. Da un punto di vista prestazionale, si evidenziano gli eccellenti risultati raggiunti da PTV.

La vera criticità di oggi invece è tenere completi ed aggiornati i dati delle mappe, partendo da quelli riguardano strettamente la cartografia, fino ai dati sulla viabilità e la percorribilità delle strade.

Senza dati al pari con le aspettative, infatti, si possono creare situazioni spiacevoli in cui i risultati trovati sono inammissibili nella realtà delle cose, come dimostra il caso di Primafrost.

Siamo sicuri che le persone di TPS hanno realizzato quanto importante sia questa priorità, e sicuramente nei prossimi anni si impegnereanno a fondo per migliorare la qualità delle loro mappe.

Questo potrebbe davvero rendere possibile, in un futuro molto prossimo, una reale disponibilità di una soluzione commerciale in grado di risolvere istanze del "Vehicle Routing Problem" in un tempo ragionevole.

Per quanto riguarda il problema trattato nella sezione e approfondito nel capitolo 3, ovvero la ricerca di coefficienti per ottenere un risultato particolare dallo strumento, si è visto che nel caso i coefficienti siano 2 il tempo impiegato risulta poco rispetto alla buona qualità della soluzione. Nel caso si renda necessario manipolare più coefficienti invece l'approccio manuale potrebbe diventare troppo lungo senza al contempo garantire nulla sull'ottimalità della soluzione. Si consiglia pertanto l'utilizzo di un algoritmo basato su metaeuristiche, come l'algoritmo genetico ivi descritto, poiché garantisce di ottenere una soluzione tutto sommato buona richiedendo un ammontare di risorse appena superiore.

Bibliografia

[1] Sito Wincor Nixdorf, sezione Retail

(http://www.wincor-nixdorf.com/internet/site_IT/IT/Retail/Retail_node.html)

[2] "The Vehicle Routing Problem", P. Toth, D. Vigo, *Society for Industrial and Apllied Mathematics*, 2002

[3] Sito TPS Italia, "Geocodifica e normalizzazione delle anagrafiche"

(http://www.tpsitalia.it/software/pianificazione_ottimizzazione_giri/geocodifica_normalizzazione_anagrafiche.php)

[4] "The Truck Dispatching Problem", Dantzig, G.B., Ramser, J.H. (1959). Management Science 6 (1): 80–91. doi:10.1287/mnsc.6.1.80. Retrieved 2008-04-17.

(http://links.jstor.org/pss/2627477)

[5] Sito PTV, sezione "Pianificazione ed ottimizzazione dei giri"

(http://www.tpsitalia.it/software/pianificazione_ottimizzazione_giri/)

[6] "MEFISTO: A Pragmatic Metaheuristic Framework for Adaptive Search with a Special Application to Pickup and Delivery Transports",A Cardeneo, W Heid, F Radaschewski et al., Operations Research Proceedings 2008: Selected Papers of the Annual International Conference of the German Operations Research Society (GOR) University of Augsburg, September 3-5, 2008

[7] "A tabu search heuristic for the vehicle routing problem", M. Gendreau, A. Hertz, G. Laporte, Management Science, 1994

[8] "The granular tabu search and its application to the Vehicle-Routing Problem", P. Toth, D. Vigo, INFORMS Journal on Computing, 2003

[9] "New insertion and postoptimization procedures for the traveling salesman problem", M. Gendreau, A. Hertz, G. Laporte, Operations Research, 1992

[10] http://www.wincor-nixdorf.com/internet/site_DE/DE/WincorNixdorf/Press/pressreleases/1998-1999/KKRundGoldmanSachs_22_10_1999.html

[11] "Wincor Nixdorf", Wikipedia tedesca
(http://de.wikipedia.org/wiki/Wincor_Nixdorf)

[12] "Wincor Nixdorf" su sito "Pubbliway, il portale per il business online",
(http://www.pubbliway.com/informatica/wincor_nixdorf_s.r.l._1121241)

[13] Sito TPS, sezione "Chi siamo"
(http://www.tpsitalia.it/azienda/)

[14] "PTV Planung Transport Verkehr" sulla Wikipedia tedesca
(http://de.wikipedia.org/wiki/PTV_Planung_Transport_Verkehr)

[15] Documentazione di PTV Intertour 5.9, fornitami da TPS Italia.

[16] Sito PTV, sezione sviluppatori, "Interactive AJAX Tutorial for PTV MobilityPlatform" (http://80.146.239.135/mp-ajax-api-samples/tutorial/tutorial.html)

[17] Wikipedia inglese, "Genetic Algorithm" (http://en.wikipedia.org/wiki/Genetic_algorithm)

[18] "The VRP Web" (http://neo.lcc.uma.es/radi-aeb/WebVRP/index.html?bibliography.html)

Ringraziamenti