Sii-Mobility

Supporto di Interoperabilità Integrato per i Servizi al Cittadino e alla Pubblica Amministrazione

Trasporti e Mobilità Terrestre, SCN_00112

Deliverable ID: DE2.2a

Titolo: Stato dell'arte nei problemi di vehicle routing.

Data corrente	15-09-2016
Versione (solo il responsabile puo' cambiare versione	0.1
Stato (draft, final)	draft
Livello di accesso (solo consorzio, pubblico)	privato
WP	2
Natura (report, report e software, report e HW..)	report
Data di consegna attesa	30-06-2016
Data di consegna effettiva	15-09-2016
Referente primario, coordinatore del documento	Paolo Cianchi
Contributor	<Nome> <Cognome> <email>, <Nome> <Cognome> <email>
Coordinatore responsabile del progetto	Paolo Nesi, UNIFI, paolo.nesi@unifi.it

Sommario

1 Executive Summary....3
2 Il problema dell'instradamento dei veicoli....3
2.1 Metodi esatti....3
2.2 Metodi non esatti....4
2.2.1 Metodi approssimati....5
2.2.2 Metodi Euristici....5
2.2.3 Metodi Metaeuristici....7
2.2.4 Considerazioni....7
2.3 Stato dell'arte....8
3 Bibliografia....9

1 Executive Summary

Questo deliverable descrive lo stato dell'arte e sulla ricerca nel campo dell'instradamento dei veicoli ed in particolare Algoritmi di instradamento (veicoli e persone).

Il problema generalmente noto con il nome di Vehicle Routing Problem (VRP) tratta la gestione di veicoli idealmente da ogni punto di vista. Questa classe di problemi comprende una casistica molto varia: questo motivo fa sì che tali problemi siano difficilmente risolubili all'ottimo. Tale problema e stato proposto in [1.]; in letteratura VRP ha numerose implicazioni pratiche, di uso in tutti i settori concernenti il trasporto di merci e persone, come ad esempio:

- raccolta di posta nelle cassette postali; servizio scuolabus;
- visite mediche a domicilio;
- visite di manutenzione preventiva; raccolta rifiuti.

L'obiettivo di questo documento è quello di presentare lo stato dell'arte attuale, mettendo in evidenza gli aspetti recenti delle ricerca e dello sviluppo nei vari settori in modo da servire come punto di partenza per lo sviluppo delle ricerche specifiche.

2 Il problema dell'instradamento dei veicoli

La letteratura sui metodi risolutivi del problema di Vehicle Routing e molto ampia e sono state proposte svariate e suggestive tecniche, al punto di rendere facile un qualsiasi tentativo di classificazione e sistematizzazione universalmente accettata. Una prima classificazione possibile e la seguente:

- metodi esatti
- metodi non esatti

La maggior parte dei metodi esatti sono basati sul modellamento matematico del problema in esame, il quale viene risolto in maniera esaustiva, fino ad arrivare all'ottimo globale. Anche un metodo di brute force può essere considerato un metodo esatto, in quanto considerando tutte le permutazioni possibili, si dimostra automaticamente che la miglior soluzione trovata e ottima.

2.1 Metodi esatti

I metodi esatti si distinguono in tre filoni principali che si differenziano l'uno dall'altro per le tecniche di base utilizzate:

- Programmazione dinamica
- Programmazione matematica
- Constraint programming

Data la complessità intrinseca del problema, l'applicabilità di questi metodi e limitata dalla memoria disponibile e dal tempo a disposizione per generare la soluzione. Premettiamo che la rinuncia all'utilizzo di metodi esatti deve essere una scelta ponderata, basata non solo sulla complessità teorica del problema (non basta, insomma, che un problema sia NP-hard), ma anche sulla effettiva praticabilità di metodi esatti (da ricercarsi eventualmente nella letteratura), al tempo di elaborazione disponibile, alle dimensioni dei problemi da risolvere. Tuttavia, spesso i metodi esatti si basano sulla formulazione di un modello di programmazione matematica del problema di ottimizzazione e la complessità del problema potrebbe rendere praticamente impossibile pervenire ad una formulazione sufficientemente accurata e maneggevole. In altri ambiti, pur disponendo di una \buona" formulazione matematica, il tempo a disposizione per la soluzione del problema non consente l'utilizzo di metodi esatti.

Come già ribadito spesso risolvere un problema di programmazione lineare intera può essere di facile; in questi casi si possono ottenere delle limitazioni superiori ed inferiori del valore

ottimo della funzione obiettivo, chiamate rispettivamente upper bound e lower bound. Questo e molto importante sia perché a volte nelle applicazioni interessa il valore ottimo della funzione obiettivo, piuttosto che la soluzione ottima del problema, in quanto conoscere tale valore e fondamentale per utilizzare algoritmi di enumerazione implicita come gli algoritmi Branch and Bound. Gli algoritmi di enumerazione implicita, tramite opportune tecniche, scartano sottoinsiemi di elementi della regione ammissibile senza dover valutare esplicitamente per ciascuno di essi la funzione obiettivo, avendo stabilito che in tali sotto-insiemi non vi possono essere soluzioni migliori rispetto alla miglior soluzione nota. Tali metodi non danno garanzia sul tempo necessario per il loro completamento.

Entrando più in dettaglio, dato un problema generico, e possibile costruire uno o più problemi detti rilassamenti che forniscono una limitazione del valore ottimo della funzione obiettivo. Siano dati i seguenti problemi di ottimizzazione:

$$P\begin{cases} min_x\, f(x) \\ x \in X_p \end{cases} \quad R\begin{cases} min_x\, g(x) \\ x \in X_r \end{cases}$$

Il problema R è un rilassato del problema P se:

$$X_r \supseteq X_p$$
$$g(x) \leq f(x), \forall x \subset X_c$$

g(x) f(x) 8x 2 X_p

Sia R un rilassato di P. Allora:

- Se R è inammissibile, P è inammissibile
- Sia $\hat{x}_p$ una soluzione ottima per P e sia $\hat{x}_r$ una soluzione ottima per R. Allora $g(\hat{x}_r) \leq f(\hat{x}_p)$
- Sia $\hat{x}_r$ una soluzione ottima per R. Se $\hat{x}_r \in X_p$ e $f(\hat{x}_r) = g(\hat{x}_r)$, allora $\hat{x}_r$ è una soluzione ottima per P.

In programmazione lineare intera esistono diversi tipi di rilassamenti. Citiamo alcuni importanti tipologie come:

- Rilassamento continuo
- Rilassamento Lagrangiano

Il rilassamento continuo si ottiene sostituendo ai vincoli di integralità delle variabili, dei semplici vincoli di non negatività, oppure eliminandoli del tutto a seconda della struttura del problema; un rilassamento continuo può essere facilmente risolto con le tecniche di soluzione della programmazione lineare, inoltre se il problema originario gode della proprietà di integralità e sufficiente a risolvere il problema.

Nel rilassamento Lagrangiano si studia la struttura del problema e si individuano i vincoli che si vogliono rilassare, i cosiddetti vincoli complicanti. Una volta individuati si eliminano tali vincoli ma si aggiunge un termine di penalizzazione alla funzione obiettivo, pesato secondo un vettore di parametri del rilassamento, detto vettore dei moltiplicatori Lagrangiani.

2.2 Metodi non esatti

I metodi non esatti possono essere classificati in vari modi in base all'approccio utilizzato. Una possibile classificazione e la seguente:

- Metodi approssimati metodi euristici
- Metodi meta-euristici

2.2.1 Metodi approssimati

I metodi approssimati basano il loro funzionamento su gli stessi modelli matematici e su gli stessi approcci utilizzati dai metodi esatti. La differenza da tali metodi sta nel fatto che non si applica una risoluzione esaustiva ma si termina l'algoritmo una volta trovata una condizione di terminazione, tipicamente una condizione sul tempo di esecuzione. Ci si accontenta di una soluzione sub-ottima dando una stima della distanza dall'ottimo fornita generalmente dai lower bound trovati per il problema in esame. Al momento, i migliori risultati in quest'area di ricerca sono dati dagli algoritmi che usano la generazione di colonne. In tali metodi il problema viene diviso in più problemi: Un problema principale, ed uno o più sotto-problemi.

2.2.2 Metodi Euristici

I metodi euristici sono metodi che trovano una soluzione attraverso una regola che viene applicata ad ogni passo dell'algoritmo e possono essere classificati in due famiglie:

- Euristiche costruttive: Costruiscono una soluzione ammissibile, la quale viene ampliata ad ogni passo dell'algoritmo;
- Euristiche di ricerca locale: A partire da una soluzione completa, si cerca di migliorarla iterativamente ad ogni passo dell'algoritmo

I metodi euristici hanno il pregio di trovare mediamente soluzioni di buona qualità senza pero nessuna garanzia di vicinanza all'ottimo.

Le euristiche costruttive determinano una soluzione ammissibile, partendo solo dai dati di ingresso del problema in esame. Caratteristica comune e l'assenza o la forte limitazione del backtracking: Si parte da una soluzione vuota e si determinano in modo iterativo nuovi elementi da aggiungere ad una soluzione, no ad arrivare ad una soluzione completa. Questo processo viene chiamato criterio di espansione. Tra i vari tipi di euristiche costruttive consideriamo ora gli algoritmi greedy.

Algoritmi greedy

L'idea di questi algoritmi è adottare un criterio di espansione basato sulla scelta più conveniente in quel momento compatibilmente con i vincoli del problema: Ad ogni iterazione viene aggiunto alla soluzione l'elemento che produce il miglioramento maggiore della funzione obiettivo. Lo schema concettuale di un algoritmo greedy e di questo tipo:

1. Inizializza la soluzione S;
2. Per ogni scelta da e ettuare:
3. Prendi la decisione piu conveniente, compatibilmente con i vincoli del problema.

La grande diffusione degli algoritmi greedy e essenzialmente dovuta ai seguenti motivi:

- L'algoritmo simula quello che sarebbe il comportamento manuale piu intuitivo nella determinazione della soluzione;
- L'implementazione dell'algoritmo risulta essere relativamente semplice;
- Il tempo di calcolo richiesto per determinare la soluzione e estremamente ridotto;
- Questi algoritmi sono utili a fornire una soluzione iniziale da integrare con algoritmi più sofisticati.

In alcuni casi gli algoritmi greedy sfruttano un ordinamento degli elementi: Gli elementi che definiscono la soluzione vengono considerati secondo tale ordine ed eventualmente inseriti nella soluzione. Spesso il criterio di ordinamento degli elementi viene aggiornato dinamicamente in modo da tener conto delle scelte fatte in precedenza. Ovviamente il continuo aggiornamento degli score degli elementi comporterà un aumento nel tempo di calcolo richiesto dall'algoritmo stesso. Generalmente gli algoritmi greedy effettuano delle scelte che rispettano sempre tutti i vincoli. Per provare a ottenere delle soluzioni diverse, e possibilmente migliori, usando la stessa procedura, si può iterare l'algoritmo utilizzando ogni volta un ordinamento diverso, ottenuto da una randomizzazione della dispatching rule. Ad

esempio, lo score potrebbe essere corretto con una componente casuale, in modo da avere la possibilità di scegliere, ad ogni passo, non l'elemento migliore, ma un elemento abbastanza buono": in questo modo si potrebbe rendere meno miope l'algoritmo e conservare alcuni elementi per passi successivi, quando le scelte diventano maggiormente critiche. In alternativa, ad ogni passo, si potrebbe considerare la scelta casuale tra i migliori n elementi residui.

Ricerca locale

Le euristiche di miglioramento effettuano delle modi che a una soluzione completa, esplorando quindi l'intorno di una soluzione, cercando di migliorare il valore della funzione obiettivo.

Consideriamo un problema di minimizzazione, e una sua soluzione ammissibile x, con associato il valore della funzione obiettivo $f(x)$. La ricerca locale consiste nel de nire un intorno di x, e nell'esplorarlo in cerca di soluzioni migliori, se ve ne sono. Se, in questo intorno di x, si scopre una soluzione y per cui $f(y) < f(x)$, allora ci si sposta da x a y e si riparte da y con l'esplorazione del suo intorno. Se invece nell' intorno di x non si scopre nessuna soluzione migliore, allora vuol dire che x e un minimo locale. Nella ricerca locale classica, arrivati in un minimo locale, l'algoritmo si ferma e restituisce questo minimo come valore di output. Ovviamente, non si ha nessuna garanzia in generale che tale valore costituisca una soluzione ottima del problema; anzi, tipicamente esso può essere anche molto distante dall'ottimo globale.

Benché ogni applicazione del concetto di ricerca locale a un problema abbia le sue peculiarità, alcuni aspetti sono abbastanza comuni a molte realizzazioni. La definizione dell'intorno e in genere connessa al concetto di perturbazione di una soluzione ammissibile, nel senso che la ricerca avviene tra le soluzioni che si ottengono perturbando una soluzione iniziale. Generalmente questi algoritmi applicati al problema di VRP sono degli edge exchange algorithm ovvero effettuano degli scambi di archi. Alcuni autori parlano di forza di un intorno [2.]. Un intorno definito in un certo modo e tanto più forte quanto più la qualità delle soluzioni prodotte dall'algoritmo e indipendente dalla bontà della soluzione di partenza. Ad esempio, 3-opt per il VRP e considerato un intorno forte (almeno per grafi fino a 50 nodi): Questo fatto implica che si può affrontare il problema senza perdere tempo a generare buone soluzioni iniziali. Anzi, conviene generare molte soluzioni iniziali casualmente, sperando così di avere un campionamento rappresentativo dell'intera regione ammissibile. Un altro aspetto caratterizzante un approccio di ricerca locale e il modo in cui deve essere esplorato l'intono di una soluzione. Due strategie antitetiche sono *first improvement* e *best improvement*. Nel primo caso l'esplorazione dell'intorno termina non appena si trova una soluzione migliore di quella corrente. Nel secondo, invece, lo si esplora comunque tutto cercando il massimo miglioramento che quell'intono consente di ottenere. In genere si preferisce il primo approccio, ma non e una regola fissa.

Metodi di decomposizione

Oltre ai metodi descritti no a questo punto esistono dei metodi basati sulla decomposizione del problema che può essere effettuata nuovamente con metodi euristici. La generazione della soluzione viene suddivisa in due fasi denominate:

- cluster: genera degli insiemi di clienti
- route: genera dei percorsi

Esistono due approcci nel utilizzare queste due fasi. Nel caso si scelga cluster first, route second si generano prima gli insiemi di clienti che saranno serviti da un certo veicolo k. Dati i cluster della prima fase, si prova a costruire dei percorsi con criteri simili agli algoritmi greedy.

Al contrario nel caso route first, cluster second inizialmente si genera un circuito (giant tour), che non rispetta i vincoli, sulla base delle distanze o del tempo di percorrenza. La seconda fase riguarda la scomposizione del giant tour al fine di soddisfare i vincoli.

Ad oggi l'approccio cluster first, route second risulta quello maggiormente seguito.

2.2.3 Metodi Metaeuristici

Negli ultimi anni hanno acquisito importanza via via maggiore alcuni approcci euristici di tipo generale, detti metaeuristiche. La struttura e l'idea di fondo di ciascuna metaeuristica sono sostanzialmente fissate, ma la realizzazione delle varie componenti dell'algoritmo dipende dai singoli problemi. I metodi metaeuristici sono metodologie generali, in cui sono definite delle componenti e le loro interazioni, al fine di pervenire ad una buona soluzione.

Molti di questi sono basati su analogie con sistemi naturali, che definiscono le componenti che devono essere specializzate per i singoli problemi. Ad esempio, nel simulated annealing si paragona il processo di soluzione di un problema di ottimizzazione combinatoria, in cui si passa iterativamente da una soluzione ammissibile a un'altra, via via migliorando la funzione obiettivo, al processo con cui un solido, raffreddandosi, si porta in configurazioni con via via minore contenuto di energia. Negli algoritmi genetici, un insieme di soluzioni ammissibili e visto come un insieme di individui di una popolazione che, accoppiandosi e combinandosi tra loro, danno vita a nuove soluzioni che, se generate in base a criteri di miglioramento della specie, possono risultare migliori di quelle da cui sono state generate.

In generale le metaeuristiche sono molto e caci ma richiedono tempi molto più elevati degli euristici tradizionali, inoltre l'applicazione a un problema di ottimizzazione combinatoria richiede una messa a punto accurata e talvolta laboriosa. Citiamo, tra le metaeuristiche più note e consolidate:

- Iterated Greedy
- Simulated Annealing
- Tabu Search
- Variable Neighborhood Search
- Algoritmi genetici
- Ant Colony Optmization

2.2.4 Considerazioni

Quando il problema e/o il contesto della soluzione non renda possibile applicare tecniche di soluzione esatte, diventa necessario fornire delle buone soluzioni ammissibili in tempi di calcolo ragionevoli. Si noti che, tipicamente, la determinazione di buone soluzioni approssimate e quello che serve nelle applicazioni reali, soprattutto se riferite a problemi di grandi dimensioni; questo e essenzialmente dovuto ad una serie di fattori:

- molti dei parametri in gioco nelle applicazioni reali sono delle stime che possono essere soggette a errore, per cui non vale la pena aspettare troppo tempo per avere una soluzione ottima;
- spesso si e interessati ad avere una possibile soluzione per il problema in esame al ne di valutare velocemente degli scenari di lavoro (contesti operativi, integrazione di algoritmi di ottimizzazione in Sistemi di Supporto alle Decisioni interattivi);

Questi aspetti spiegano perché, nelle applicazioni reali, sia così di uso il ricorso a metodi che permettono di trovare delle \buone" soluzioni senza garantire la loro ottimalità ma garantendo un tempo di calcolo relativamente breve. Per la maggior parte dei problemi di ottimizzazione

combinatoria e possibile progettare delle euristiche specifiche che sfruttano le proprietà del particolare problema in esame e le conoscenze specifiche che derivano dall'esperienza.

2.3 Stato dell'arte

Il Vehicle Routing Problem (VRP) è una classe di problemi molto importanti nell'ambito della ricerca operativa e dell'ottimizzazione combinatoria e per questo e stato oggetto di numerosi studi. L'area di maggiore interesse riguarda la soluzione ottima o sub-ottima per istanze di problema a larga scala. La versione più semplice di questa classe, il Traveling Salesman Problem (TSP), puo essere risolto ottimamente con istanze con migliaia di nodi [3.], mentre istanze di VRP con centinaia di clienti, iniziano a essere difficili da risolvere ottimamente. Per questo i metodi esatti rimangono con nati alla soluzione di istanze di dimensione limitata. Per questo motivo in letteratura vengono proposti algoritmi metaeuristici, ricerche locali parallele con schema di cooperazione, approcci ibridi che stanno a metà strada tra metodi euristici e metodi esatti, e sono in grado di produrre soluzioni di alta qualità in tempi ragionevoli per le applicazioni pratiche. Nonostante i contributi e il progresso del settore, negli ultimi anni stanno emergendo molte competizioni in questo ambito. L'efficienza, la scalabilità e la rapidità delle procedure sono attributi prerequisito per il loro adattamento all'interno dei sistemi di supporto alle decisioni della vita reale, in cui i tempi di risposta devono essere ragionevolmente brevi.

Recentemente, e stata dedicata grande attenzione all'analisi di grandi insiemi di dati, di dimensioni diverse, in varianti del VRP che imitano alcune condizioni di vita reali. Gli articoli [4.] e [5.] riportano risultati per il VRP capacitivo considerando rispettivamente 1200 e 24000 clienti con approccio metaeuristico. Per VRP con flotta eterogenea [4.] conduce esperimenti con più di 480 clienti, mentre per il VRPTW [6.] hanno valutato delle istanze a larga scala con cardinali da 200 a 1000 clienti. Il VRPTW è certamente la variante più studiata dei VRPs e può essere considerato il prototipo dei cosiddetti rich VRPs [7.] a causa dei vincoli di tempo che richiedono tecniche sofisticate per verificare l'ammissibilità [8.]. Questa classe si verifica in diversi sistemi logistici di trasporto e può essere usato per modellare numerose applicazioni della vita reale [9.]. Tipicamente le applicazioni sul VRPTW di grandi dimensioni si possono trovare nella raccolta dei rifiuti [10.], fast food routing [11.], scuola bus routing [12.], consegna a domicilio [13.]. Si può anche fare riferimento a [14.] per una raccolta di applicazioni del VRP riguardo il trasporto di rifiuti solidi, di bevande, cibo, latte e la consegna a domicilio di giornali.

I primi studi sul VRPTW risalgono agli anni '60. Per una panoramica sugli sviluppi, ci riferiamo a [15.], [16.], [17.], e [18.]

In studi più recenti [19.] e [20.] esaminano gli sviluppi algoritmici nel campo delle euristiche costruttive e di miglioramento iterativo, delle metaeuristiche e degli algoritmi evolutivi.

Il Multi Depot Vehicle Routing Problem with Time Windows (MDVRPTW) ha come riferimento dello stato dell'arte [21.]. In questo articolo si propongono due schemi di cooperazione per comporre una variante parallela della metaeuristica Variable Neighborhood Search (VNS). Uno schema detto a grana larga, designato per trovare e memorizzare soluzioni, che verranno poi migliorate, e con un meccanismo di auto-tuning dei parametri. Un altro schema, detto a grana ne, progettato per riprodurre le caratteristiche di successo della VNS sequenziale. La combinazione con l'uso parallelo di questi due approcci, ha trovato tutte le miglior soluzioni migliorandone 11 su 20 per le istanze proposte in [22.].

3 Bibliografia

[1.] Dantzig and Ramster. Optimum routing of gasoline delivery trucks. 1959.

[2.] Thomas Stutzle. Iterated local search-variable neighborhood search, 2003.

[3.] Gutin and A. Punnen. The traveling salesman problem and its variations. Kluwer Academic Publishers, 2002.

[4.] F. Li, B.L Golden, and E.A. Wasil. Very large-scale vehicle routing: New test problems, algorithms, and results. Computers & Operations Research 32, 2005.

[5.] J. Kytöjoki, T. Nuortio, O. Bräysy, and M. Gendreau. An effcient variable neighborhood search heuristic for very large scale vehicle routing problems. Computers & Operations Research, 2007.

[6.] H. Gehring and J. Homberger. A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows. 1999.

[7.] M. Gendreau and C.D. Tarantilis. Solving large-scale vehicle routing problems with time windows: The state-of-the-art. 2010.

[8.] S. Irnich. A unified modeling and solution framework for vehicle routing and local search-based metaheuristics. Journal of Computing, 2008.

[9.] L. Bodin, B. Golden, A. Assad, and M. Ball. Routing and scheduling of vehicles and crews: The state-of-the-art. Computers & Operations Research, 1983.

[10.] S. Sahoo, S. Kim, B.-I. Kim, B. Kraas, and A.Jr. Popov. Routing optimization for waste management. 2005.

[11.] R.A. Russell. Hybrid heuristics for the vehicle routing problem with time windows. 1995.

[12.] J. Bracca, J. Bramel, and D. Simchi-Levi. A computerized approach to the new york city school bus routing problem. 1994.

[13.] D. Weigel and B. Cao. Applying gis and or techniques to solve sears technician-dispatching and home-delivery problems. 1999.

[14.] B.L. Golden, A.A. Assad, and E.A. Wasil. Routing vehicles in the real world: Applications in the solid waste, beverage, food, dairy, and newspaper industries. 2002

[15.] B.L. Golden and A.A. Assad. Perspectives on vehicle routing: Exciting new developments. Operations Research, 1986.

[16.] M. Desrochers, J.K. Lenstra, M.W.P. Savelsbergh, and F. Soumis. Vehicle routing with time windows: Optimization and approximation. 1988.

[17.] Desrosiers, Y. Dumas, M.M. Solomon, and F. Soumis. Time constrained routing and scheduling. 1995.

[18.] J.-F. Cordeau, G. Desaulniers, J. Desrosiers, M.M. Solomon, and F. Soumis. The vrp with time windows. 2002.

[19.] Braysy O. and Gendreau M. Vehicle routing problem with time windows part i: Route construction and local search algorithms. 2005.

[20.] J.-F. Cordeau, G. Desaulniers, J. Desrosiers, M.M. Solomon, and F. Soumis. The vrp with time windows. 2002.

[21.] M. Polacek, S. Benkner, K.F. Doerner, and R.F. Hartl. A cooperative and adaptive variable neighborhood search for the multi depot vehicle routing problem with time windows. 2008.

[22.] J.-F. Cordeau, G. Laporte, and A. Mercier. A unified tabu search heuristic for vehicle routing problems with time windows. Journal of the Operational Research Society, 2001.