

Metodi e Modelli per l'Ottimizzazione Combinatoria

Progetto – Descrizione di dettaglio dei problemi

Luigi De Giovanni

Trovate di seguito la descrizione formale del problema proposto per il progetto. La descrizione è piuttosto dettagliata, perché prova a rispondere a varie possibili ambiguità messe in luce da una o più delle vostre proposte (qualcosa potrà sembrarvi ovvio, ma per qualcuno di voi non lo era).

ATTENZIONE!

La **prossima scadenza** è relativa alla consegna di un modello di programmazione lineare intera mista che risponda a queste specifiche (e non a quelle che ciascuno di voi aveva formulato).

Il termine è fissato per **lunedì 19 novembre, ore 9:00 AM**.

Nota: non sono vietati (anzi sono incoraggiati) la ricerca e l'adattamento di modelli che si trovino nella letteratura scientifica (ad esempio, si può copiare da un articolo scientifico, non dal collega...). In ogni caso, dopo la seconda consegna, il docente metterà a disposizione dei modelli di riferimento per tutti, da implementare con Cplex.

1 Configurazione di reti wireless ad hoc

Alcuni dispositivi dotati di sistemi di comunicazione wireless si trovano in una certa area e vogliono stabilire una connessione, finalizzata alla conduzione di un gioco che coinvolga tutti i dispositivi.

1.1 Connessione con un solo server

Il modello di connessione prevede l'elezione tra i dispositivi di un server con i seguenti compiti:

- raccolta di informazioni sullo stato di tutti i dispositivi;
- elaborazione delle informazioni raccolte;
- distribuzione delle informazioni aggiornate a tutti i dispositivi.

Tutti i restanti dispositivi agiscono da client: elaborano strategie di gioco e comunicano informazioni di stato al server a intervalli di tempo regolari. Anche il server partecipa attivamente al gioco e, quindi, elabora le relative strategie.

Per ciascun dispositivo (detto anche nodo) v è nota la posizione in termini di coordinate cartesiane X_v e Y_v , e la distanza massima R_v (raggio di trasmissione) entro la quale deve trovarsi un altro dispositivo per poter stabilire una connessione (link). Pertanto, dati due nodi a e b , esiste il link (cioè la possibilità di stabilire una connessione) solo se la distanza tra a e b è inferiore o uguale sia a R_a che a R_b . Posizione e raggio di trasmissione di ogni nodo rimangono invariati per tutta la durata del gioco e sono dati dal committente.

Qualora esistessero uno o più nodi isolati, cioè nodi che non possono stabilire nessun link con altri dispositivi, il gioco non potrebbe avere inizio (il problema non avrebbe soluzione, risulterebbe cioè inammissibile).

Le connessioni da e per il server possono essere multi-hop (ad esempio se non c'è un link diretto tra client e server, o per altre esigenze operative): in questo caso, i dispositivi intermedi dovranno ricevere e ritrasmettere le informazioni di altri dispositivi inoltrandole verso la destinazione. La comunicazione delle informazioni sullo stato del gioco avviene punto-punto, la modalità di trasmissione è unicast e non è previsto multi-cast: il server genera dei pacchetti diversi per ciascun client, anche se i pacchetti dovessero contenere la stessa informazione (in realtà ogni client riceve informazioni specializzate, quindi le informazioni sono generalmente diverse). In particolare:

- ogni T_C unità di tempo, ciascun client v invia al server un pacchetto di dati di dimensione D_v^C ;

- ogni T_S unità di tempo, il server invia a ciascun client v un pacchetto di dati di dimensione D_v^S .

I parametri T_C, T_S sono costanti per l'intera durata del gioco e sono dati in unità di tempo (ad esempio [s = secondi]). Per ogni nodo v , D_v^C e D_v^S sono dati in unità di informazione (ad esempio [B = byte]).

All'inizio, ciascun nodo v è dotato di una carica nota A_v (che può essere diversa per ciascun dispositivo) espressa in unità di carica (ad esempio [J = joule]), che viene consumata durante il gioco per i seguenti motivi:

- per i client:
 - consumi della CPU per l'elaborazione delle strategie di gioco;
 - consumi per la trasmissione delle informazioni di stato al server;
 - consumi per la ricezione delle informazioni di stato dal server;
 - eventuali consumi per la ricezione e la ritrasmissione di informazioni di stato provenienti dal server o da altri client.
- per il server:
 - consumi della CPU per l'elaborazione delle strategie di gioco (come client);
 - consumi della CPU per l'elaborazione delle informazioni di stato raccolte;
 - consumi per la ricezione delle informazioni di stato dai client;
 - consumi per la trasmissione delle informazioni di stato aggiornate ai client.

Per la quantificazione dei consumi valgono le seguenti osservazioni:

- i consumi per l'elaborazione delle strategie di gioco da parte del client v (incluso il server visto come client) sono costanti nel tempo e dipendono dal nodo client: tali consumi sono indicati con P_v^C e sono dati dal committente in termini di unità di carica per unità di tempo (ad esempio [J/s]);
- i consumi del server per l'elaborazione delle informazioni di stato raccolte dipendono dal nodo v che funge da server: sono indicati con P_v^S e sono dati dal committente in termini di unità di carica per unità di tempo (ad esempio [J/s]);
- è possibile calcolare i consumi in unità di carica (ad esempio [J]) sostenuti dal nodo trasmittente i per la trasmissione (o la ritrasmissione) di un pacchetto verso il nodo ricevente j usando la seguente formula:

$$M_{ij}^T \cdot b + C_{ij}^T$$

dove b è la dimensione del pacchetto in unità di informazione (ad esempio [B]), M_{ij}^T è il consumo per unità di informazione trasmessa (espresso in unità di carica per unità

di informazione, ad esempio $[J/B]$) e C_{ij}^T è il consumo fisso per pacchetto (espresso in unità di carica, ad esempio $[J]$). I parametri M_{ij}^T e C_{ij}^T sono costanti per tutta la durata del gioco e sono dati dal committente;

- è possibile calcolare i consumi in unità di carica (ad esempio $[J]$) sostenuti dal nodo ricevente j per la ricezione di un pacchetto proveniente dal nodo trasmittente i usando la seguente formula:

$$M_{ij}^R \cdot b + C_{ij}^R$$

dove b è la dimensione del pacchetto in unità di informazione (ad esempio $[B]$), M_{ij}^R è il consumo per unità di informazione ricevuta (espresso in unità di carica per unità di informazione, ad esempio $[J/B]$) e C_{ij}^R è il consumo fisso per pacchetto (espresso in unità di carica, ad esempio $[J]$). I parametri M_{ij}^R e C_{ij}^R sono costanti per tutta la durata del gioco e sono dati dal committente.

Si fa notare che i parametri per la quantificazione dei consumi di trasmissione e ricezione dipendono dal link (e non solo dal trasmettitore o ricevitore) in quanto le caratteristiche del link potrebbero richiedere modalità di trasmissione diverse (ad esempio un aumento di potenza): tali stime sono lasciate al committente.

Oltre che richiedere il consumo di energia dei nodi, le trasmissioni richiedono disponibilità di banda sui link utilizzati. Tale disponibilità potrebbe dipendere dalla distanza tra i nodi, o dalle caratteristiche tecnologiche dei dispositivi interessati, o da altri parametri in possesso del committente. In prima approssimazione, la banda disponibile sul link si assume non dipenda dal numero di connessioni stabilite dai nodi interessati. In particolare, per ogni coppia ordinata di nodi (i, j) il committente fornisce la capacità U_{ij} , parametro che definisce la velocità massima delle trasmissioni da i a j , ossia la banda massima in unità di informazione per unità di tempo (ad esempio $[B/s]$). Il dato potrebbe essere fornito in modo indiretto, ad esempio attraverso formule in funzione di parametri come quelli sopra indicati e comunque forniti dal committente. Se il link (i, j) è disponibile, secondo le regole sopra definite, allora la sua capacità di banda massima è U_{ij} . *Si fa notare che la disponibilità di ciascun link e la relativa banda disponibile sono definibili a priori, in base alle regole riportate.* Qualora uno o più link non fossero in grado di sopportare tutte le richieste di trasmissione, e non ci fossero altri link disponibili, il gioco non potrebbe svolgersi correttamente (in questo caso il problema non ha soluzione, risulta cioè inammissibile).

Il flusso di informazione da un nodo client v al nodo server s è *splittable*, cioè può essere suddiviso su più percorsi da v a s . Questo vuol dire che pacchetti diversi originati in v potrebbero seguire ciascuno un percorso (successione di link) diverso (mentre lo stesso pacchetto rimane indivisibile). Inoltre, le comunicazioni da v a s e da s a v possono avvenire su percorsi diversi, non necessariamente simmetrici: ad esempio, se la trasmissione dal nodo 1 al nodo 4 passano in successione dai nodi $1 - 2 - 3 - 4$, le trasmissioni da 4 a 1 potrebbero utilizzare percorsi diversi da $4 - 3 - 2 - 1$. Le stesse osservazioni

valgono per i flussi di informazione dal server verso ciascun client. Si noti che la possibilità di suddividere i pacchetti su percorsi diversi può essere utile per bilanciare i consumi di trasmissione su diversi nodi. Pertanto, indicando con s il server scelto e con v un qualsiasi altro nodo client, per ogni coppia di nodi (v, s) (risp. (s, v)), l'instradamento (routing) dei pacchetti è definito in termini di:

- insieme dei percorsi (successione di link) da v a s (risp. da s a v) utilizzati per la trasmissione,
- percentuale dei pacchetti su ciascun percorso.

Il routing è stabilito all'inizio ed è mantenuto per l'intera durata del gioco.

La connessione tra i dispositivi e il gioco si considerano attivi finché tutti i dispositivi sono accesi o, in altri termini, finché uno qualsiasi dei dispositivi esaurisce la batteria; a quel punto il gioco termina, a prescindere da quale dispositivo abbia terminato la sua scorta di energia.

Si assume che:

- i parametri T_C e T_S indichino intervalli di tempo di durata trascurabile rispetto alla durata del gioco;
- la durata delle diverse trasmissioni sia trascurabile rispetto alla durata del gioco: la trasmissione di può sempre considerare istantanea, indipendentemente dalla qualità del segnale sul link o sui link interessati;
- i consumi dei nodi per attività diverse da quelle sopra descritte sono trascurabili.

Si vuole determinare la configurazione della rete in termini di:

- scelta del nodo che svolge la funzione di server,
- routing dei pacchetti da ciascun client verso il server scelto (percorsi e percentuali di pacchetti su ciascun percorso),
- routing dei pacchetti dal server scelto verso ciascun client (percorsi e percentuale di pacchetti su ciascun percorso),

al fine di massimizzare la durata del gioco.

1.2 Connessione con server dormienti

Per cercare di aumentare la durata del gioco, è possibile considerare un diverso modello di connessione in cui più di un dispositivo, a turno, funge da server. In questo caso, invece di scegliere un solo server, si sceglieranno un numero fissato L di nodi e, in ciascun turno, uno solo dei nodi configurati come server è attivo, mentre gli altri sono dormienti. A intervalli regolari di durata T_H data, il server attivo diventa dormiente e un server dormiente diventa attivo, secondo una regola round robin. Il parametro T_H è fornito dal committente e rimane costante per l'intera durata del gioco. Inoltre, la durata T_H si considera trascurabile rispetto alla durata del gioco: di conseguenza possiamo assumere che ciascun server dormiente ricopra il ruolo di server attivo per un tempo pari a $1/L$ del tempo totale di durata del gioco (l'ordine di avvicendamento non è pertanto rilevante, in quanto alla fine del gioco, il tempo per cui ciascun server dormiente sarà stato effettivamente attivo sarà lo stesso per tutti a meno della costante T^H , che ha valore trascurabile).

I nodi client (così come il server attivo e i server dormienti, che si comportano anche come client), come nel modello precedente, elaborano strategie di gioco e inviano / ricevono le informazioni sullo stato del gioco al / dal solo server attivo.

Il server attivo, oltre ai compiti descritti in precedenza, si occupa di comunicare, a intervalli regolari, informazioni di sincronizzazione ai server dormienti (queste informazioni includono, al momento opportuno, anche il comando di cambio turno, che quindi non costituiscono traffico aggiuntivo). In particolare, ogni T_D unità di tempo, il server attivo invia a ciascun server dormiente v un pacchetto di sincronizzazione di dimensione D_v^D . I parametri T_D e D_v^D sono costanti per l'intera durata del gioco e sono dati, rispettivamente, in unità di tempo [s] e in unità di informazione [B]. Si noti che i pacchetti potrebbero essere diversi per i diversi server dormienti e che, come per le altre comunicazioni, sono trasmessi in modalità unicast (una trasmissione per ogni server dormiente, senza multicast).

Pertanto, oltre ai consumi sopra elencati, si devono considerare:

- consumi del server attivo per la trasmissione delle informazioni di sincronizzazione ai server dormienti;
- consumi dei server dormienti per la ricezione delle informazioni di sincronizzazione;
- consumi dei server dormienti per l'elaborazione delle informazioni di sincronizzazione;
- i consumi dei client per la ricezione e la ritrasmissione delle informazioni di sincronizzazione dirette ai server dormienti.

Per la quantificazione dei consumi descritti nella sezione precedente, vale quanto già detto.

Per la quantificazione dei consumi ulteriori dovuti alle informazioni di sincronizzazione¹, valgono le seguenti osservazioni:

- i consumi dei server dormienti per l'elaborazione delle informazioni di sincronizzazione dipendono dal nodo v che si vedrà assegnato il ruolo di server dormiente: sono indicati con P_v^D , costanti per l'intera durata del gioco e sono dati dal committeente in termini di unità di carica per unità di tempo (ad esempio [J/s]);
- è possibile calcolare i consumi in unità di carica (ad esempio [J]) per la trasmissione (o ritrasmissione) e la ricezione dei pacchetti di sincronizzazione secondo le formule specificate sopra, tenendo conto che la dimensione b dei pacchetti di sincronizzazione è data dal parametro D_v^D .

Per quanto riguarda l'instradamento dei pacchetti, vale quanto detto sopra circa la possibilità di splitting dei flussi. Inoltre, si deve tener conto che ogni volta che cambia il server attivo, si adotta un diverso instradamento (in termini di percorsi e percentuale di pacchetti su ciascun percorso), sempre stabilito all'inizio e mantenuto per l'intera durata del gioco.

Analogamente a quanto visto sopra, si assume che:

- il parametro T_D indichi un intervallo di tempo di durata trascurabile rispetto alla durata del gioco;
- la durata delle trasmissioni di sincronizzazione sia trascurabile rispetto alla durata del gioco: la trasmissione di può sempre considerare istantanea, indipendentemente dalla qualità del segnale sul link o sui link interessati;
- i consumi dei nodi per attività diverse da quelle sopra descritte sono trascurabili.

Si vuole determinare la configurazione della rete in termini di:

- scelta dei nodi che svolgeranno la funzione di server (si ricorda che, secondo quanto detto sopra, ciascun server sarà attivo per $1/L$ della durata del gioco e dormiente per $(L - 1)/L$ della durata del gioco, e che l'ordine di avvicendamento round-robin non è rilevante),
- routing dei pacchetti da ciascun client verso ciascuno dei server scelti (per ciascun server, percorsi e percentuali di pacchetti su ciascun percorso, da utilizzare nel turno relativo al server stesso),

¹Anche se la considerazione non fa parte delle specifiche, si fa notare che, nonostante ci siano dei consumi aggiuntivi rispetto alla configurazione con un solo server, la possibilità di distribuire su più nodi le funzioni del server e i relativi consumi potrebbe far complessivamente aumentare la durata del gioco.

- routing dei pacchetti da ciascuno dei server scelti verso ciascun client (per ciascun server, percorsi e percentuali di pacchetti su ciascun percorso, da utilizzare nel turno relativo al server stesso),
- routing dei pacchetti da ciascuno dei server scelti quando è attivo verso ciascun altro server quando è dormiente (per ciascun server, percorsi e percentuali di pacchetti su ciascun percorso, da utilizzare nel turno relativo al server stesso),

al fine di massimizzare la durata del gioco.

1.3 Riepilogo dei dati forniti dal committente

Si fornisce l'elenco dei parametri forniti dal committente. Si assumeranno, per semplicità espositiva, le seguenti unità di misura:

- unità di tempo: secondi [s];
- unità di informazione: byte [B]
- unità di carica: joule [J]

1.3.1 Connessione con un solo server

I parametri necessari per descrivere un'istanza del problema di configurazione di reti wireless ad hoc con modello di connessione a un solo server sono:

- N : insieme dei dispositivi (detti anche nodi);
- X_v e Y_v , per ogni $v \in N$: coordinate in un sistema cartesiano bidimensionale della posizione del nodo v [metri];
- R_v , per ogni $v \in N$: raggio di trasmissione del nodo v [metri];
- A_v , per ogni $v \in N$: carica iniziale del nodo v [J];
- T_C : intervallo di tempo tra due trasmissioni di pacchetti di stato da un client al server [s];
- D_v^C , per ogni $v \in N$: dimensione del pacchetto con le informazioni di stato spedite dal nodo v [B];
- T_S : intervallo di tempo tra due trasmissioni di pacchetti di aggiornamento dal server a ciascun client [s];
- D_v^S , per ogni $v \in N$: dimensione del pacchetto con le informazioni di aggiornamento destinate al nodo v [B];
- P_v^C , per ogni $v \in N$: consumo per unità di tempo per le elaborazioni di v come client [J/s];

- P_v^S , per ogni $v \in N$: consumo aggiuntivo per unità di tempo per le elaborazioni di v come server [J/s];

Inoltre, indicando con E l'insieme dei link determinabile in base ai parametri X_v , Y_v e R_v , il committente fornisce²:

- M_{ij}^T , per ogni $(i, j) \in E$: consumi per unità di informazione trasmessa sul link (i, j) [J/B];
- C_{ij}^T , per ogni $(i, j) \in E$: consumi per pacchetto trasmesso sul link (i, j) [J];
- M_{ij}^R , per ogni $(i, j) \in E$: consumi per unità di informazione ricevuta sul link (i, j) [j/B];
- C_{ij}^R , per ogni $(i, j) \in E$: consumi per pacchetto ricevuto sul link (i, j) [J];
- U_{ij} , per ogni $(i, j) \in E$: banda disponibile sul link (i, j) [B/s].

1.3.2 Connessione con server dormienti

Nel caso si voglia determinare la configurazione con l'utilizzo di server dormienti, il committente fornisce:

- tutti i parametri per il caso di connessione con un solo server;
- L : numero di server che si alternano in round-robin [numero intero];
- T_H : intervallo di tempo tra due cambiamenti di server attivo [s];
- T_D : intervallo tra due trasmissioni di pacchetti di sincronizzazione dal server attivo a ciascuno dei server dormienti [s];
- D_v^D , per ogni $v \in N$: dimensione del pacchetto con le informazioni di sincronizzazione inviate dal server attivo al nodo v configurato come server dormiente [B];
- P_v^D , per ogni $v \in N$: consumo aggiuntivo per unità di tempo per le elaborazioni del nodo v come server dormiente [J/s].

²Il committente, in base a quanto sopra esposto, potrebbe fornire gli stessi dati per ciascuna coppia ordinata di nodi $(i, j) \in N \times N$, in modo diretto o indiretto (tramite formule o regole per ricavarli). In ogni caso, i dati forniti sono da applicarsi ai soli link $(i, j) \in E$.

2 Un esempio

Si descrive di seguito un esempio di istanza del problema, con, in molti casi, informazioni semplificate.

Connessione con un solo server

- $N = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$;
- dispositivi disposti a griglia con distanza orizzontale e verticale di 40 metri
 $(X_1, Y_1) = (0, 0)$, $(X_2, Y_2) = (40, 0)$, $(X_3, Y_3) = (80, 0)$
 $(X_4, Y_4) = (0, 40)$, $(X_5, Y_5) = (40, 40)$, $(X_6, Y_6) = (80, 40)$
 $(X_7, Y_7) = (0, 80)$, $(X_8, Y_8) = (40, 80)$, $(X_9, Y_9) = (80, 80)$;
- $R_v = 50$ metri, per tutti i nodi $v \in N$;
- $A_v = 9000$ J, per tutti i nodi $v \in N$;
- $T_C = 0,011$ s;
- $D_v^C = 84$ B, per tutti i nodi $v \in N$;
- $T_S = 0,050$ s;
- $D_v^S = 184$ B, per tutti i nodi $v \in N$;
- $P_v^C = 1,0$ J/s, per tutti i nodi $v \in N$;
- $P_v^S = 0,3$ J/s, per tutti i nodi $v \in N$;
- $M_{ij}^T = 0,48 \times 10^{-6}$ J/B, per tutti i link $(i, j) \in E$;
- $C_{ij}^T = 431 \times 10^{-6}$ J, per tutti i link $(i, j) \in E$;
- $M_{ij}^R = 0,12 \times 10^{-6}$ J/B, per tutti i link $(i, j) \in E$;
- $C_{ij}^R = 316 \times 10^{-6}$ J, per tutti i link $(i, j) \in E$;
- $U_{ij} = 2,5 \times 10^6$ B/s, per tutti i link $(i, j) \in E$.

Dati aggiuntivi per il caso di connessione con server dormienti

- $L = 3$;
- $T_H = 10$ s;
- $T_D = 0,050$ s;
- $D_v^D = 500$ B, per tutti i nodi $v \in N$;
- $P_v^D = 0,05$ J/s, per tutti i nodi $v \in N$.

3 Osservazioni e suggerimenti per la modellazione

Al fine della formulazione di un modello matematico del problema potrebbero essere utili le seguenti osservazioni.

Attenzione! *Alcune osservazioni sono relative a una particolare formulazione del problema, che non è l'unica possibile e che potrebbe essere diversa da quella che ciascuno svilupperà.*

- Le diverse comunicazioni richieste (da ciascun client al server, dal server a ciascun client, dal server attivo a ciascun server dormiente) sono assimilabili a diverse commodity che devono essere instradate sulla rete.
- Il routing è stato definito, in sede di specifica, in termini di percorsi utilizzati e quantità del flusso su ciascun percorso. Il routing può essere definito in modo equivalente in termini di link, fornendo, per ciascuna commodity, la quantità di flusso che passa su ciascun link. In questo modo le variabili che descrivono il routing sono quelle che abbiamo visto per il modello di flusso multicommodity, del tipo x_{ij}^k : quantità di flusso della commodity k instradata sul link (i, j) .
- Come detto nella specifica, anche se la durata del gioco non è nota, visto che è oggetto di ottimizzazione, possiamo comunque assumere che i tempi T_S , T_C e T_D siano trascurabili rispetto alla durata del gioco stessa. Inoltre, il routing rimane fissato per tutta la durata del gioco. Pertanto possiamo, senza perdere precisione, considerare il comportamento medio della rete, e, in particolare, modellare i flussi in trasmissione, ritrasmissione o ricezione in termini di *pacchetti nell'unità di tempo*, in modo da poter calcolare i relativi consumi in termini di carica nell'unità di tempo (si fa notare che i consumi di CPU sono già forniti in J/s).

Ad esempio, con riferimento ai parametri sopra quantificati, dire che un client invia un pacchetto al server attivo ogni 0,011 s, equivale a dire che sono inviati mediamente $1/0,011 = 90,909$ pacchetti al secondo.

In questo modo, un possibile modello potrebbe determinare i consumi per unità di tempo per le comunicazioni di ciascun dispositivo.

Ad esempio, la sola trasmissione dei pacchetti di stato verso il server, comporta per un client un consumo di $90,909 \times (0,480 \times 84 + 431) \times 10^{-6}$ J/s = 0,043 J/s

Sommando i consumi per unità di tempo dovuti alle comunicazioni e alle elaborazioni della CPU, si può ricavare il consumo complessivo per unità di tempo di ciascun dispositivo e, da questo, ricavare la durata di ciascun dispositivo (e, quindi, massimizzare la minima durata).

Ad esempio, un client “periferico”, che non ritrasmette traffico per altri client o server dormienti, ha i seguenti consumi:

- *elaborazione*: 1 J/s ;
- *invio pacchetti al server*: $0,043 \text{ J/s}$;
- *ricezione pacchetti dal server*: $1/0,050 \times (0,120 \times 184 + 316) \times 10^{-6} \text{ J/s} = 0,007 \text{ J/s}$;

per un totale di $1,05 \text{ J/s}$. Pertanto, la batteria con 9000 J di carica iniziale permette una durata del client di $9000/1,05 = 8571.43 \text{ s}$.

- Come detto nel punto precedente, possiamo modellare i flussi considerando il numero medio di pacchetti per unità di tempo. Pertanto, eventuali variabili associate al numero di pacchetti trasmessi, ritrasmessi o ricevuti per unità di tempo (corrispondenti alle x_{ij}^k viste per il modello di flusso multicommodity) sono variabili continue.