

Appunti paper scientifici MCS

Game theory in MCS: A Comprehensive Survey

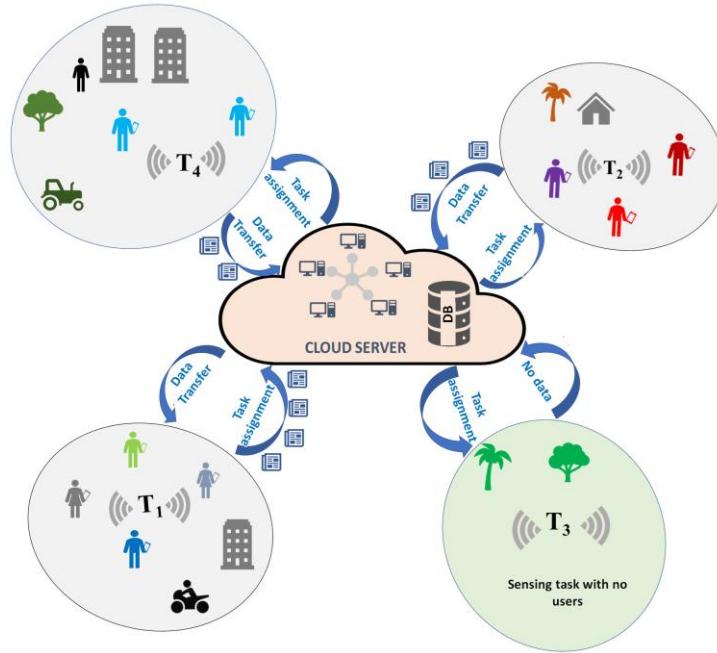
Abstract

MCS è un paradigma emergente nell'acquisizione distribuita di dati per smart city e Internet of Things (IoT). Il MCS richiede un gran numero di utenti per consentire l'accesso ai sensori integrati nei loro dispositivi mobili e condividere i dati rilevati per garantire un elevato valore e un'alta veridicità dei dati massivi rilevati. Migliorare la partecipazione degli utenti nelle campagne MCS richiede di motivare efficacemente gli utenti, il che è una preoccupazione fondamentale per il successo delle piattaforme MCS.

Questo articolo presenta una revisione completa delle diverse soluzioni basate sulla teoria dei giochi che affrontano le seguenti problematiche nel MCS, come il costo di rilevamento, la qualità dei dati, la determinazione del prezzo ottimale tra richiedenti e fornitori di dati e gli incentivi. È stata proposta una tassonomia delle soluzioni basate sulla teoria dei giochi per le piattaforme MCS in cui i problemi sono principalmente formulati sulla base dei giochi di **Stackelberg**, **Bayesiani** ed **Evolutivi**. Il criterio iniziale per categorizzare le soluzioni basate sulla teoria dei giochi per il MCS si basa sulla cooperazione e sulle informazioni disponibili tra i partecipanti, mentre un partecipante potrebbe essere un richiedente o un fornitore.

Introduzione

L'acquisizione dei dati mediante i sensori è fondamentale per una piattaforma MCS; difatti, entra in gioco il **rilevamento partecipativo o opportunistico**: sfruttando questi sensori, è possibile raccogliere enormi volumi di dati e processarli in tempo reale. Pertanto, MCS permette agli utenti con dispositivi intelligenti (banalmente smartphone o smartwatch) di partecipare attivamente o passivamente alla raccolta di dati, ricompensandoli per il loro contributo. Nelle campagne MCS, i dati vengono raccolti da diverse posizioni utilizzando i sensori integrati nei dispositivi mobili, reclutando gli utenti sia in modo **implicito (opportunistico)** che **esplicito (partecipativo)** (come illustrato qui sotto in figura):



I dati acquisiti vengono poi aggregati, elaborati, analizzati e visualizzati per supportare una varietà di servizi che contribuiscono a creare spazi più intelligenti e sostenibili.

Nonostante questi vantaggi, la costruzione di un sistema MCS è una sfida complessa. Ad esempio, la **garanzia di affidabilità e veridicità** dei dati raccolti dagli utenti è un ostacolo enorme poiché coinvolge fonti anonime e spesso non del tutto affidabili; dunque, la mancanza di fiducia nei dati può derivare da **manipolazioni intenzionali (azioni malevoli di avversari)** o anche da malfunzionamenti temporanei dei sensori. L'unico strumento a disposizione per proteggersi da queste vulnerabilità è un'efficace selezione degli utenti: ciò significa progettare sistemi che gestiscano e aggiornino la reputazione degli utenti, adottando regole o politiche di reclutamento ottimali.

Altra problematica è la **privacy**, che può scoraggiare gli utenti al partecipare alle campagne MCS: per superare questo potenziale ostacolo, sono stati proposti metodi efficaci di incentivazione (sia monetari o meno). Dunque, è innegabile che i costi di rilevamento, elaborazione e trasmissione dati sostenuti durante le campagne MCS richiedano una forma di compensazione.

Un sistema MCS necessita anche di una **diversità di partecipanti** al fine di garantire che i dati sensoriali combinati siano i più accurati possibili rispetto al valore reale, migliorando così la qualità complessiva dei dati crowd-sensed. Da questa [fonte](#), è stato esaminato il comportamento dei partecipanti in base a diverse ricompense monetarie, scoprendo che stimolare la competizione tra i partecipanti potrebbe non aiutare a ottenere informazioni di alta qualità tramite dati raccolti in modo collaborativo; invece, è stato riportato che una progettazione adeguata dei micropagamenti può portare a una partecipazione più utile.

Un'ultima questione importante dal punto di vista dell'utente è il **consumo energetico**. Si prevede che in un futuro prossimo sarà possibile alimentare i dispositivi intelligenti tramite segnali wireless. Questa innovazione potrebbe ridurre drasticamente i costi di rilevamento e facilitare il processo di raccolta dati, incoraggiando un maggior numero di utenti a partecipare alle campagne di rilevamento senza preoccuparsi della batteria. Inoltre, la **mobilità degli utenti** nel MCS migliora l'area di copertura del rilevamento.

Per affrontare tutti questi obiettivi contrastanti, diverse ricerche recenti propongono l'adozione di approcci basati sulla **Teoria dei Giochi**, dove ogni partecipante mira a massimizzare il proprio profitto, ovvero la funzione obiettivo di ogni giocatore. La teoria dei giochi analizza situazioni in cui sono coinvolti più giocatori e il guadagno di un giocatore non dipende solo dalla propria strategia. Nel MCS, il gioco può essere formulato tra i partecipanti stessi o tra la piattaforma e i partecipanti. Nel primo caso, i partecipanti al Crowdsensing competono tra loro per ottenere un profitto maggiore e la decisione di ogni partecipante sarà influenzata dalle azioni degli altri. Nel secondo caso, le decisioni della piattaforma sugli incentivi agli utenti sono influenzate dalle strategie dei partecipanti. In entrambi i casi, il "profitto" può manifestarsi sotto forma di massimizzazione del punteggio, della reputazione o dei benefici ottenuti dagli utenti.

Contesto e sfide nel MCS

Mantenere un'elevata **qualità dei dati** nel Mobile Crowdsensing (MCS) è complicato per diverse ragioni. Anzitutto, problemi come i **malfunzionamenti dei sensori** o le **condizioni ambientali** possono compromettere l'accuracy delle rilevazioni. Inoltre, per coprire aree vaste e raccogliere un volume significativo di dati, è necessario **reclutare molti partecipanti**. Questo si traduce in un **aumento dei costi** per la piattaforma MCS, poiché gli utenti si aspettano, giustamente, un incentivo per il loro contributo. Paradossalmente, però, raccogliere dati oltre una certa quantità **non garantisce automaticamente un miglioramento della qualità**. Per ovviare a questo, una possibile soluzione è adottare **strategie di stima** per le aree non coperte dai partecipanti. Ad esempio, si possono utilizzare i valori delle "celle" vicine per inferire dati mancanti. Tuttavia, questo approccio presenta un rischio: se i dati stimati non sono accurati, la **qualità complessiva si degrada**, portando a **decisioni errate**.

In questa [fonte](#) sono presenti diversi meccanismi di assegnazione task per vari scenari per migliorare la qualità dei dati: è stato dimostrato che la qualità dei dati è direttamente associata alla durata della task, ai tempi di raccolta dei dati e alla copertura spazio-temporale. D'altra parte, l'inferenza dei dati mancanti dovrebbe essere eseguita senza alcun errore sul server per soddisfare i requisiti di qualità: le tecniche di ricampionamento come il **bootstrapping**, ad esempio, potrebbero essere utilizzate per approssimare la qualità a valori ragionevoli.

Inoltre, nel caso di rilevamento partecipativo nelle campagne MCS, la qualità dei dati può variare a causa della **mancanza di esperienza nei partecipanti** (ad esempio, catturare foto sfocate, registrare audio con microfono coperto, ecc.). Oltre a tutto ciò, la potenza di calcolo e le capacità sensoriali di ogni smartphone variano, quindi non tutti i partecipanti potrebbero essere in grado di fornire lo stesso livello di qualità dei dati rilevati. Poiché la maggior parte dei meccanismi di stima della qualità non tratta gli errori dei sensori e l'affidabilità dell'utente in modo diverso, questa ingenua assunzione di non disaccoppiare utente e smartphone in termini di affidabilità può portare a una valutazione imprecisa dei partecipanti durante il processo di reclutamento. Per affrontare questa sfida, è stato [proposto](#) un algoritmo che sfrutta la matrice di confusione e separa gli errori dell'utente e del dispositivo per quantificare la qualità di un partecipante.

L'affidabilità gioca un ruolo fondamentale poiché i dati ottenuti dai partecipanti influenzano il processo decisionale. La fiducia può essere stabilita sotto forma di **fiducia per affidabilità** o **fiducia per decisione**:

- La prima viene calcolata su un'entità e denota la probabilità con cui tale entità riporta la verità;
- La seconda si riferisce alla situazione in cui un'entità è considerata affidabile indipendentemente dai suoi possibili errori.

Un problema di una piattaforma MCS riguarda il **livello di affidabilità sui meccanismi di sicurezza**; in particolare: *in che misura il meccanismo di sicurezza può essere considerato affidabile?* Per rispondere a questa domanda, è stata utilizzata la [funzione di Gompertz](#) per progettare un sistema di reputazione che garantisce l'affidabilità nei dati ricevuti.

Un approccio innovativo per migliorare l'affidabilità consiste nell'utilizzare **punteggi di reputazione collaborativi**. Questo significa combinare la **reputazione statistica** (basata su dati e misurazioni oggettive) con la **reputazione basata sul voto** ([ottenuta tramite le valutazioni di altri utenti](#)), il tutto supportato da un'assistenza di ancoraggio per validare ulteriormente queste valutazioni.

La regolazione del consumo energetico ha una preoccupazione quasi uguale nel MCS con la protezione della sicurezza e della privacy degli utenti. Il rilevamento, l'elaborazione dei dati grezzi e il trasferimento dei dati raccolti richiedono energia della batteria. La riduzione del livello della batteria di un dispositivo dipende da molti fattori come il sistema operativo (OS) utilizzato dal dispositivo, il tipo e il numero di sensori utilizzati per il rilevamento e inoltre il GPS consuma una quantità diversa di energia a seconda che il dispositivo sia al chiuso o all'aperto. Il consumo energetico e la velocità di trasferimento dei dati variano a seconda del tipo di modalità di rete scelta (3G, WiFi, LTE, ecc.) e della frequenza di campionamento. In questo [paper](#), è stato proposto uno schema di gestione energetica distribuita in cui gli utenti vengono penalizzati per l'utilizzo di elevata energia, anche se potrebbero mantenere la qualità dei dati. Inoltre, può essere utile considerare il livello della batteria residuo dei dispositivi mobili per reclutare gli utenti, poiché la bassa energia residua è uno dei motivi per cui si rinuncia a una campagna di rilevamento; tuttavia, ciò si potrebbe tradurre in una bassa **Qol (Quality of Information)**. Al fine di acquisire una migliore Qol, è stato proposto un modello di selezione consapevole dell'energia in cui la frequenza di campionamento degli utenti variava in base al livello della batteria in modo da consentire alla piattaforma di reclutare utenti con batteria scarica per soddisfare la qualità delle informazioni richiesta.

I partecipanti di una piattaforma MCS usano le proprie risorse — tempo, energia e denaro — per raccogliere e inviare dati. È naturale aspettarsi un qualche tipo di beneficio in cambio. Per questo, tale piattaforma deve essere progettata per **motivare** le persone a partecipare tramite **incentivi** poiché operano con **budget fissi**. Non tutti i partecipanti dovrebbero ricevere lo stesso incentivo, dato che la **qualità del servizio** o il **contributo** possono variare significativamente da utente a utente.

Da ciò scaturisce la seguente domanda: **come si valuta la qualità dei dati forniti da un utente quando non esiste un riferimento certo (ground truth)?** In questo [documento](#) è stato proposto di stimare la qualità dei dati e il livello di contributo combinando la **teoria dell'informazione** con l'algoritmo **Expectation-Maximization (EM)**.

È importante notare che diverse task hanno priorità diverse: alcuni richiedono una **qualità dei dati** impeccabile, altri puntano a una **copertura geografica** più ampia. Meccanismi di incentivazione ben studiati possono gestire queste diverse esigenze:

- Ad esempio, per task che richiedono alta qualità si potrebbero incentivare gli utenti in base alla loro **reputazione**;
- Invece per task che necessitano di ampia copertura la **disponibilità** dell'utente diventa un fattore chiave.

Per attirare gli utenti in un luogo di rilevamento, gli incentivi offerti non dovrebbero essere inferiori ai costi effettivi di rilevamento; questo squilibrio di prezzo è una delle sfide presentate in [un'indagine](#) dettagliata sui meccanismi di incentivo nel MCS, come ad esempio reclutare partecipanti con budget inferiori attraverso **l'asta inversa**.

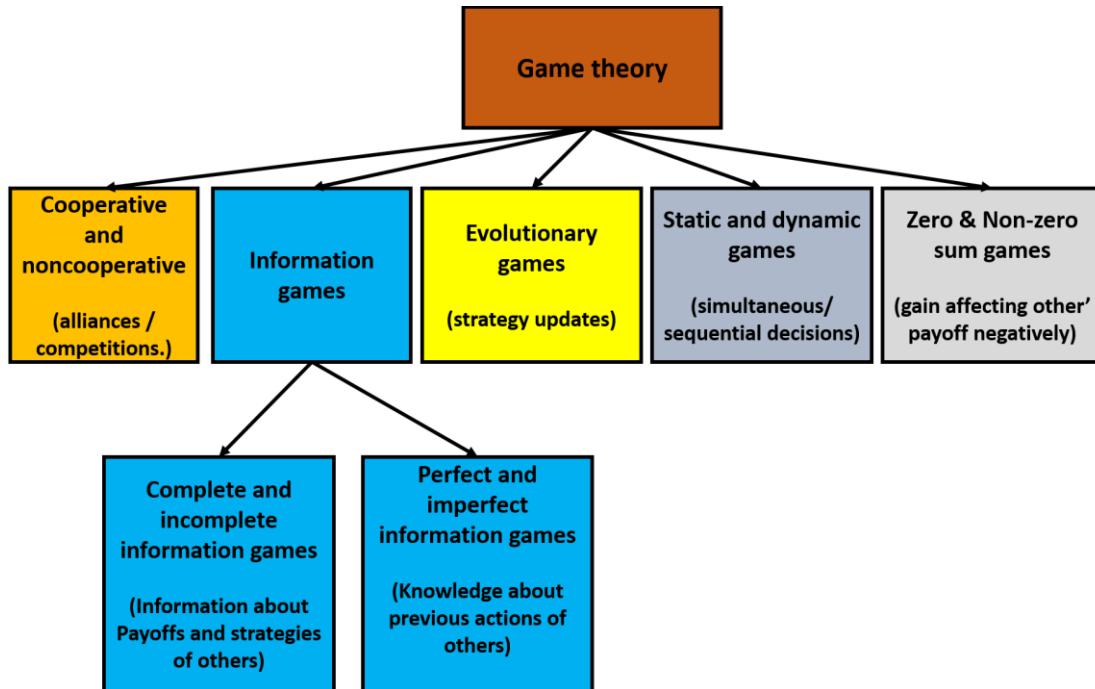
Introduzione ai modelli comuni della Teoria dei Giochi

La **teoria dei giochi** è uno strumento utile per modellare il comportamento dei partecipanti al fine di massimizzare le loro [funzioni di utilità](#). La ragione principale alla base della formulazione di queste interazioni sotto forma di giochi deriva dall'egoismo e dalla razionalità dei partecipanti. Pertanto, in presenza di utenti egoisti e razionali, la cooperazione dei partecipanti può essere garantita attraverso giochi efficaci. Un caso d'uso tipico che sfrutta la teoria dei giochi è la progettazione di incentivi per reclutare una popolazione di partecipanti sufficientemente numerosa per le [campagne di rilevamento](#); inoltre, le strategie di prezzo per il valore dei dati, al fine di eliminare i partecipanti non veritieri nelle campagne MCS, servono a risolvere i problemi di sicurezza e affidabilità nelle campagne MCS.

Date le diverse limitazioni e problematiche da valutare riguardo gli utenti e i loro dispositivi mobili, come ad esempio il consumo energetico, la teoria dei giochi funge da strumento utile per raggiungere un ragionevole compromesso tra l'utilità della piattaforma MCS e il consumo energetico dei dispositivi partecipanti; inoltre, è fondamentale massimizzare i requisiti di privacy degli utenti.

Le tre componenti principali che costituiscono i modelli di gioco sono i **giocatori**, le **strategie** e i **guadagni**. Un giocatore rappresenta un'entità che prende decisioni, mentre una strategia rappresenta un insieme di regole per le decisioni da compiere in varie situazioni e il guadagno può essere una ricompensa o una perdita che i giocatori sperimentano quando implementano le loro rispettive strategie.

La seguente figura illustra i modelli popolari nelle soluzioni di teoria dei giochi, suddivisi in categorie e sottocategorie:



Giochi cooperativi e non cooperativi

Un gioco è detto **cooperativo** quando due o più giocatori cooperano tra loro per ricevere un guadagno migliore. In questa modalità, i giocatori ricevono assistenza da terze parti per abilitare la cooperazione garantendo sanzioni e incentivi, evitando di colpire coloro che dovrebbero essere esonerati al fine di evitare qualsiasi disonestà nella partecipazione. Tendenzialmente, i partecipanti mirano a obiettivi e interessi comuni che consentono loro di stabilire: la cooperazione avrà successo quando il guadagno ottenuto è considerevolmente superiore agli sforzi individuali dei giocatori. D'altra parte, un gioco è detto **non cooperativo** se i giocatori non possono formare gruppi necessari per abilitare la cooperazione. Ogni giocatore viene trattato come un'unità a sé stante poiché si presuppone che ogni utente agisca in base al proprio interesse.

Giochi a informazione

Nei **giochi a informazione**, gli utenti possono avere una visione perfetta o meno riguardo le decisioni prese dagli altri utenti a un gioco oppure se sono dotati o meno della conoscenza delle possibili strategie e guadagni. Da qui nascono diverse tipologie e forme di tale modalità.

- **Gioco a informazione perfetta:** i partecipanti hanno conoscenza delle decisioni precedenti prese dagli altri partecipanti del gioco (il gioco del tris ne è un esempio). Tipicamente, un giocatore compie un'azione dopo aver osservato la decisione di un suo avversario e viceversa;
- **Gioco a informazione imperfetta:** costituisce la conoscenza comune delle possibili strategie del tipo di giocatore ma non delle azioni scelte dagli altri;
- **Gioco a informazione completa:** i partecipanti possono essere dotati della conoscenza delle possibili strategie e dei guadagni degli altri;
- **Gioco a informazione incompleta:** i partecipanti potrebbero non avere una parziale o totale conoscenza relativi ai guadagni e alle strategie adottate. Questa modalità può essere suddivisa ulteriormente in due categorie:
 - **Simmetrico:** è un caso in cui tutti i giocatori hanno la stessa quantità di informazioni private, ma nessuno conosce completamente le strategie o i guadagni degli altri. In altre parole, l'incertezza è distribuita equamente tra tutti i partecipanti;
 - **Asimmetrico:** l'opposto di simmetrico.

Giochi statici e dinamici

Un **gioco statico** si riferisce a situazioni in cui due o più giocatori prendono le loro decisioni in contemporanea, senza sapere cosa faranno gli altri (un esempio è il *dilemma del prigioniero*). Questi giochi vengono spesso visualizzati in una tabella (una matrice) che elenca i partecipanti, le loro possibili strategie e i guadagni (o le perdite) che ogni giocatore ottiene a seconda delle combinazioni di scelte.

Invece, un **gioco dinamico** si riferisce a situazioni in cui due o più giocatori prendono le loro decisioni che si susseguono nel tempo, una dopo l'altra, oppure sono giochi che si ripetono. Un esempio è la *dama*, dove ciascun giocatore decide la sua mossa basandosi su quella appena fatta dall'avversario. Le scelte dei giocatori sono sempre influenzate dalle informazioni disponibili in quel momento.

Giochi a somma zero e a somma diversa da zero

Nei **giochi a somma zero**, il guadagno di un partecipante corrisponde esattamente alla perdita degli altri facendo sì che la somma complessiva dei guadagni e delle perdite sia sempre pari a zero; dunque, l'interesse di un utente è completamente opposto a quello degli altri (scacchi è un esempio, dove c'è solamente un vincitore).

Al contrario, nei **giochi a somma diversa da zero**, il totale dei profitti e delle perdite dei partecipanti non è né zero né una somma costante. Questo tipo di giochi permette strategie più complesse, spesso definite **strategie miste**. In questi scenari, i partecipanti non sono né completamente in opposizione né totalmente collaborativi. Possono competere parzialmente e allo stesso tempo cooperare per ottenere risultati migliori per tutti. Ciò significa che il guadagno di un giocatore non si traduce necessariamente in una perdita per gli altri.

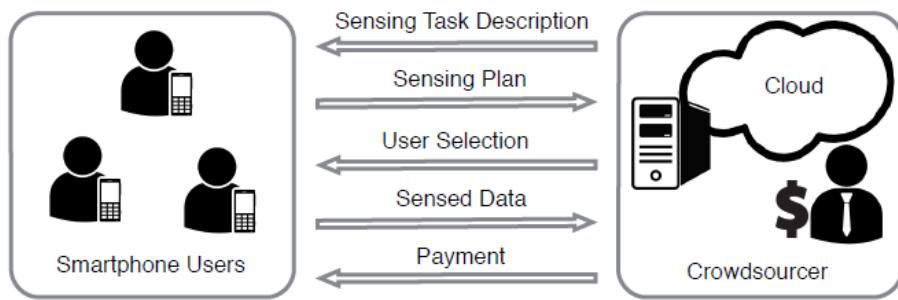
Incentive Mechanisms for Crowdsensing: Crowdsourcing with Smartphones

Abstract

Grazie ai sensori integrati di uno smartphone, come giroscopio, GPS, microfono, fotocamera e tanti altri, è possibile raccogliere una grande mole di dati da diverse attività svolte dagli utenti (su ciò che li circonda o meno). Con un sistema crowdsensing, il **crowdsourcer** può reclutare diverse tipologie di utenti per ottimizzare i propri sistemi mediante un processo di assegnazione dei task e di incentivazione degli utenti, con partecipazione attiva o passiva. Tipicamente, una buona piattaforma MCS dovrebbe essere dotata di buoni meccanismi di incentivazione al fine di attrarre più utenti per la partecipazione alle attività da svolgere.

Esistono due tipologie di modelli MCS: un modello **gestito dal crowdsourcer**, in cui quest'ultimo fornisce una ricompensa condivisa agli utenti che partecipano alla piattaforma, e un modello **incentrato sull'utente**, dove i partecipanti hanno più controllo riguardo i pagamenti che riceveranno (es. attraverso meccanismi di offerta o negoziazione):

- Nel primo modello, è fondamentale la realizzazione di un meccanismo di incentivazione basato sul **gioco di Stackelberg**, dove viene calcolato un *equilibrio unico di Stackelberg* la cui funzione di utilità del crowdsourcer è massimizzata e nessun utente può migliorare la sua utilità deviando unilateralmente dalla sua strategia attuale.
- Nel secondo modello, è fondamentale la realizzazione di un meccanismo di incentivazione basato su **asta** come il **modello di Vickrey** e deve essere efficiente dal punto di vista computazionale, razionale a livello individuale, redditizio e veritiero (**truthful**).



In questo contesto, viene realizzato un sistema composto da un *crowdsourcer* che risiede in un sistema cloud ed è composto da diversi server di rilevamento, e da numeri utenti collegati tramite cloud. Inizialmente, il crowdsourcer rende pubbliche diverse task di rilevamento. Supponiamo che un insieme di utenti $U = \{1, 2, \dots, n\}$ sia interessato a partecipare presso la piattaforma MCS dopo aver letto la descrizione del task interessato, con $n \geq 2$.

Quando un utente si unisce a questa piattaforma di **crowdsensing**, ci rimette qualcosa, ad esempio tempo, batteria del telefono o dati mobili: per compensare questo "costo", si aspetta di essere **pagato** per il servizio che offre. Così, ogni utente decide il proprio *piano strategico*: può indicare quanto tempo è disposto a dedicare al rilevamento dati, oppure specificare il prezzo minimo a cui è disposto a vendere le informazioni raccolte. Questo piano viene poi inviato al **crowdsourcer**, calcola quanto spetta a ciascuno e invia i **pagamenti**. A quel punto, solo gli utenti scelti eseguono effettivamente le attività di rilevamento e inviano i dati raccolti al crowdsourcer, completando così il ciclo.

Il crowdsourcer è interessato unicamente a massimizzare la propria utilità. Poiché gli smartphone sono di proprietà di individui diversi, è ragionevole assumere che gli utenti siano **egoisti ma razionali**; di conseguenza, ogni utente desidera massimizzare la propria utilità e non parteciperà al crowdsensing a meno che non ci sia un incentivo sufficiente. L'obiettivo di questo articolo è la progettazione di **meccanismi di incentivazione** che siano semplici, scalabili e con proprietà probabilmente buone. Altri aspetti della progettazione e implementazione dell'intero sistema di crowdsensing sono al di fuori dello scopo di questo articolo.

I modelli di riferimento per una piattaforma MCS sono i seguenti:

- **Modello crowdsourcer-centrico:** in questo modello, il piano strategico di un utente interessato si basa sul tempo che dedica all'attività. Un utente che partecipa al crowdsensing riceverà un pagamento non inferiore al suo costo; tuttavia, dovrà competere con altri utenti per una **somma totale di pagamento fissa**.
- **Modello utente-centrico:** in questo modello, ogni utente stabilisce un prezzo per il proprio servizio: se selezionato, l'utente riceverà un pagamento non inferiore al prezzo richiesto. A differenza del modello *crowdsourcer-centrico*, il pagamento totale non è fisso per il modello utente-centrico; quindi, gli utenti hanno più controllo sul compenso in quest'ultimo modello.

Funzionamento del modello crowdsourcer-centrico

In questo contesto, analizziamo il caso in cui abbiamo solamente un task da svolgere e il crowdsourcer mette “sul piatto” una ricompensa totale $R > 0$ per spingere un certo numero di utenti n a partecipare. Ogni utente, dal canto suo, decide quanto vuole impegnarsi in base a questa ricompensa. La strategia di ciascun utente viene indicata con $t_i \geq 0$, che indica semplicemente quanto tempo è disposto a dedicare al rilevamento dei dati. Partendo dal presupposto che tutti gli utenti che partecipano siano disponibili per l'intera durata del task e che ogni minuto dedicato abbia lo stesso valore per il crowdsourcer, ciascun utente ha un costo da affrontare calcolato come $k_i t_i$, dove $k_i \in \Theta$ è il costo unitario dell'utente e Θ rappresenta l'insieme di tutti i possibili costi unitari di tutti gli utenti.

Questo insieme di costi unitari Θ è a disposizione del crowdsourcer e sa anche qual è la distribuzione di probabilità degli utenti in base al loro costo unitario (ad esempio, quanti utenti hanno un costo unitario Θ_j): queste informazioni possono essere ricavata mediante analisi dei dati.

Assumiamo che n_j rappresenti il numero di utenti che hanno costo unitario Θ_j ; dunque, il numero totale di utenti presenti nella piattaforma è:

$$n = \sum_{\Theta_j \in \Theta} n_j$$

Supponendo che la ricompensa che deve ricevere l'utente i sia proporzionale al tempo/strategia che dedica, allora la sua funzione di utilità (che rappresenta il suo guadagno netto) è data dalla ricompensa meno il costo:

$$\bar{u}_i = \frac{t_i}{\sum_{j \in U} t_j} (R - k_i t_i)$$

N.B.: È importante specificare che questo modello non si applica direttamente se i costi non sono lineari. Un punto chiave è che, in base a questa funzione di utilità, tutti gli utenti che hanno lo stesso costo unitario tenderanno a scegliere lo stesso piano di rilevamento.

Invece, la funzione di utilità del crowdsourcer è la seguente:

$$\bar{u}_0 = g(\tilde{t}_1, \tilde{t}_2, \dots, \tilde{t}_l; n_1, n_2, \dots, n_l) - R$$

dove \tilde{t}_j rappresenta il tempo di rilevamento scelto dagli utenti con costo unitario k_j , mentre la funzione $g(\dots, \dots)$ rappresenta quanto il crowdsourcer valuta il tempo totale di rilevamento offerto dagli utenti. Se avessimo una situazione in cui $g(0, \dots, 0; 0, \dots, 0) = 0$, significa che nessun utente desidera partecipare e quindi la funzione di utilità del crowdsourcer è nulla. In generale, questa funzione g è strettamente concava rispetto ai tempi di rilevamento e aumenta monotonicamente con l'aumentare di ciascun tempo di rilevamento \tilde{t}_j .

L'obiettivo è identificare il valore ottimale della ricompensa R che massimizzi la funzione di utilità del crowdsourcer \bar{u}_0 ; allo stesso tempo, ciascun utente sceglierà il proprio t_i per massimizzare la propria funzione di

utilità in base al valore di R . Logicamente, nessun utente razionale accetterebbe di lavorare per un'utilità negativa, quindi l'utente i non parteciperà ($t_i = 0$) se la ricompensa offerta è troppo bassa rispetto al suo costo, ovvero quando:

$$R \leq k_i \sum_{j \neq i \in \mathcal{U}} t_j$$

Funzionamento del modello utente-centrico

In questo modello, il *crowdsourcer* annuncia un insieme di task $\Gamma = \{\tau_1, \tau_2, \dots, \tau_m\}$ di task che gli utenti possono selezionare, dove ogni singola task ha un valore $v_j > 0$ e la selezione dei task dipende essenzialmente dalle preferenze; di conseguenza, l'utente i ha un costo associato c_i , che è privato e noto solo a sé stesso. L'utente i quindi sottopone al *crowdsourcer* la coppia (Γ_i, b_i) , dove b_i rappresenta il prezzo di riserva al quale l'utente desidera vendere il proprio servizio.

Una volta ricevute tutte le coppie da tutti gli utenti, il *crowdsourcer* seleziona un sottoinsieme di utenti \mathcal{S} come idonei e determina il pagamento p_i per ogni utente i . La funzione di utilità dell'utente i è la seguente:

$$\tilde{u}_i = \begin{cases} p_i - c_i, & \text{if } i \in \mathcal{S}, \\ 0, & \text{otherwise.} \end{cases}$$

Invece, la funzione di utilità del *crowdsourcer* è la seguente:

$$\widetilde{u_0} = v(\mathcal{S}) - \sum_{i \in \mathcal{S}} p_i = \sum_{\tau_j \in \cup_{i \in \mathcal{S}} \Gamma_i} - \sum_{i \in \mathcal{S}} p_i$$

dove l'operatore unione serve per evitare possibili duplicazioni.

Per il modello centrato sull'utente, miriamo a progettare un meccanismo di incentivazione che soddisfi le seguenti quattro proprietà desiderabili:

- **Efficienza Computazionale:** Il risultato dell'asta può essere calcolato in tempo polinomiale;
- **Razionalità Individuale:** Ogni utente partecipante avrà una funzione di utilità non negativa, sottomettendo la sua vera valutazione;
- **Redditività:** Il valore apportato dai vincitori non dovrebbe essere inferiore al pagamento totale loro corrisposto;
- **Veridicità:** Nessun offerente può migliorare la propria utilità sottomettendo un'offerta diversa dalla sua vera valutazione (che in questo articolo è il costo), indipendentemente da ciò che gli altri sottomettano.

L'importanza delle prime tre proprietà è ovvia, perché esse insieme assicurano la fattibilità del meccanismo di incentivazione. Essendo veritiero, il meccanismo di incentivazione può eliminare la paura della manipolazione del mercato e il sovraccarico di strategie sugli altri per gli utenti partecipanti.

Meccanismo di incentivazione per un modello crowdsourcer-centrico

Immaginiamo un sistema in cui un crowdsourcer vuole che un gruppo di utenti S esegua un task, come ad esempio la raccolta dei dati tramite sensori: per incentivare questi utenti, il crowdsourcer deve offrire una ricompensa R . Questo scenario può essere modellato come un **gioco di Stackelberg**, che in questo caso prende il nome di **IMCC (Incentive Mechanism for Crowdsourcing)**. Tipicamente, in un gioco di Stackelberg c'è un leader e uno o più follower: il leader prende la sua decisione per primo e i follower rispondono a questa decisione. Nel caso del gioco IMCC, il leader deve annunciare una ricompensa R e i follower decidono quanto tempo (t_i) dedicare al task (τ_j).

Questo processo si svolgerà in due fasi distinte:

1. Il crowdsourcer annuncia pubblicamente la ricompensa R che intende offrire per il lavoro svolto;
2. Dopo l'annuncio di R , ciascun utente deciderà autonomamente quanto tempo t_i dedicare al task, cercando di massimizzare la propria funzione di utilità \bar{u}_i ovvero il proprio guadagno netto.

Indichiamo il tempo/strategia di tutti gli utenti (profilo strategico) con $t = (t_1, t_2, \dots, t_n)$, dove n è il numero totale di utenti; inoltre, se consideriamo l'utente i , indichiamo con t_{-i} il profilo strategico di tutti gli utenti tranne quella di i e quindi possiamo scrivere $t = (t_i, t_{-i})$. **L'obiettivo del modello IMCC** è fare in modo che il crowdsourcer sia interessato unicamente alla massimizzazione della propria funzione di utilità e non quella collettiva (degli utenti); inoltre, è fondamentale assumere che il numero di utenti sia molto elevato poiché la strategia di un singolo utente ha un impatto trascurabile sulla strategia complessiva.

La seconda fase di IMCC, quella in cui gli utenti decidono i loro tempi di rilevamento, può essere vista come un **gioco non cooperativo** a sé stante che chiameremo **Determinazione del Tempo di Rilevamento (SDT)**. Date queste premesse, le domande fondamentali che ci poniamo sono:

- **Q1: Esistenza di Strategie Stabili:** Data una ricompensa R annunciata dal crowdsourcer, esiste un insieme di strategie $t = (t_1, t_2, \dots, t_n)$ nel gioco STD tale che nessun utente avrebbe un vantaggio nel cambiare unilateralmente la propria decisione sul tempo di rilevamento?
- **Q2: Unicità delle Strategie Stabili:** Se la risposta a Q1 è sì, questo insieme di strategie stabili è unico? L'unicità è cruciale perché permetterebbe al crowdsourcer di prevedere con certezza il comportamento degli utenti. Se la soluzione è unica, tutti gli utenti convergeranno verso lo stesso comportamento prevedibile.
- **Q3: Ottimizzazione della Ricompensa del Crowdsourcer:** Come può il crowdsourcer scegliere il valore ottimale di R per massimizzare la propria utilità, definita da $\bar{u}_0 = g(\tilde{t}_1, \tilde{t}_2, \dots, \tilde{t}_l; n_1, n_2, \dots, n_l) - R$?

La strategia menzionata nella domanda Q1 corrisponde esattamente al concetto di **Equilibrio di Nash (NE)**.

Definizione 1 (Equilibrio di Nash): Un insieme di strategie $t_1^{ne}, t_2^{ne}, \dots, t_n^{ne}$ costituisce un Equilibrio di Nash del gioco STD se, per ogni utente i , la sua funzione di utilità risulta massimizzata scegliendo la strategia t_i^{ne} e non può essere ulteriormente migliorata modificandola unilateralmente, a condizione che tutti gli altri utenti mantengano invariate le proprie strategie di equilibrio.

Formalmente, questa definizione può essere espressa come segue:

$$\bar{u}_i(t_i^{ne}, t_{-i}^{ne}) \geq \bar{u}_i(t_i, t_{-i}^{ne}), \quad \forall t_i \geq 0$$

L'**Equilibrio di Nash (NE)** è cruciale perché rappresenta una situazione stabile: nessun giocatore ha motivo di cambiare la propria strategia se gli altri mantengono le loro. Al contrario, una situazione senza un NE è instabile. L'**unicità dell'NE** è particolarmente utile: se NE è unico, il crowdsourcer può prevedere come si comporteranno gli utenti e, di conseguenza, scegliere il valore migliore per una variabile chiave (R). Per questo motivo, la soluzione al problema Q3 è strettamente legata alle risposte ai problemi Q1 e Q2; inoltre, la soluzione ottimale che si ottiene per Q3, combinata con l'NE di STD, forma quella che viene definita un **Equilibrio di Stackelberg** per IMCC.

Determinazione del piano strategico di un utente

Innanzitutto, è fondamentale definire il concetto di **strategia di risposta migliore**:

Definizione 2 (Strategia di risposta migliore): Data la strategia scelta da tutti gli utenti tranne l'utente i (indicata con t_{-i}), la strategia di miglior risposta dell'utente i è quella scelta t_i , indicata con $\beta_i(t_{-i})$, che massimizza la sua utilità $\bar{u}_i(t_i, t_{-i})$ con $t_i \geq 0$.

In base alla **Definizione 1**, ciascun utente applica una propria strategia al fine di fornire la migliore risposta; in particolare, dalla funzione di utilità \bar{u}_i sappiamo che il $t_i \leq \frac{R}{k_i}$ poiché avremmo $\bar{u}_i < 0$ e ciò non avrebbe senso.

Studiando la migliore strategia di un utente i , possiamo calcolare le derivate parziali della funzione di utilità rispetto alla sua strategia:

$$\frac{\partial u_i}{\partial t_i} = \frac{-Rt_i}{(\sum_{j \in U} t_j)^2} + \frac{R}{\sum_{j \in U} t_j} - k_i \Rightarrow \frac{\partial^2 \bar{u}_i}{\partial t_i^2} = -\frac{2R \sum_{j \in U_{-i}} t_j}{(\sum_{j \in U} t_j)^3} < 0$$

Poiché la derivata seconda della funzione di utilità è negativa, allora possiamo dire che è strettamente concava rispetto a t_i ; dunque, per ogni $R > 0$ e per ogni profilo di strategia t_{-i} degli altri utenti, la strategia di miglior risposta $\beta_i(t_{-i})$ dell'utente i è unica se esiste. Inoltre, se la strategia di tutti gli altri utenti $j \neq i$ è $t_j = 0$, allora l'utente i non ha una strategia di miglior risposta poiché può ottenere una funzione di utilità arbitrariamente vicina a R scegliendo t_i sufficientemente piccolo ma positivo; perciò, è fondamentale considerare solamente il caso in cui $\sum_{j \in U_{-i}} t_j > 0$. Imponendo la derivata prima della funzione di utilità dell'utente i pari a 0, avremo che:

$$\frac{-Rt_i}{(\sum_{j \in U} t_j)^2} + \frac{R}{\sum_{j \in U} t_j} - k_i = 0 \Rightarrow t_i = \sqrt{\frac{R \sum_{j \in U_{-i}} t_j}{k_i}} - \sum_{j \in U_{-i}} t_j$$

Se la strategia adottata da un utente porta a un valore strettamente positivo (ovvero, **maggiore di 0**), allora la **concavità della sua funzione di utilità** garantisce che esista una **migliore risposta unica e ottimale** da parte di tale utente. Al contrario, se il valore della strategia è **minore o uguale a 0**, l'utente non avrà alcun incentivo a partecipare all'attività proposta. E quindi, avremo che la migliore risposta fornita dall'utente ha questi due casi:

$$\beta_i(t_{-i}) = \begin{cases} 0, & \text{if } R \leq \kappa_i \sum_{j \in U_{-i}} t_j; \\ \sqrt{\frac{R \sum_{j \in U_{-i}} t_j}{\kappa_i}} - \sum_{j \in U_{-i}} t_j, & \text{otherwise.} \end{cases}$$

L'algoritmo per determinare il calcolo dell'equilibrio di Nash è il seguente:

Algoritmo 5 → Calcolo dell'equilibrio di Nash nel contesto di una piattaforma MCS

- Ordino i costi unitari degli utenti in un set Θ ($\theta_1 < \theta_2 < \dots < \theta_l$);
- Costruisco un set di costi unitari vuoto, detto come Θ_w : esso conterrà i costi unitari degli utenti che parteciperanno all'attività MCS;
- Si assume che $j \in [1, l]$ sia il più piccolo valore tale che $\sum_{k=1}^j n_k \geq 2 \rightarrow$ si assume, dunque, che ci siano almeno due utenti idonei;
- Incrementando il valore di j di 1, itero per ciascun utente idoneo al fine di determinare quali classi di costo unitario faranno parte del set Θ_w ; infatti, se il costo unitario θ_j risulta essere più piccola di questa condizione $\frac{(\theta_j + \sum_{k=1}^{j-1} n_k \theta_k)}{\sum_{k=1}^{j-1} n_k}$, allora bisognerà verificare se esiste almeno un utente con quel costo così da aggiungerlo nel set Θ_w e per poi incrementare j di 1 → questo passo è utile per selezionare gli utenti in base alle loro efficienze;
- Si determinano il numero totale di utenti selezionati dal crowdsourcing $n_0 = \sum_{\theta_k \in \Theta_w} n_k$, e per ciascun costo unitario $\theta_j \in \Theta_w$ si determina il tempo/strategia ottimale di ciascun utente $\tilde{t}_j = \frac{(n_0 - 1) \cdot R}{\sum_{\theta_k \in \Theta_w} n_k \theta_k} \cdot \left(1 - \frac{(n_0 - 1) \cdot \theta_j}{\sum_{\theta_k \in \Theta_w} n_k \theta_k} \right)$
- Si realizza un insieme di utenti idonei, che conterrà gli indici le cui unità di costo appartengono a $\Theta_w \rightarrow S = \{i | \kappa_i \in \Theta_w\}$
- Se l'utente i si trova in S , ovvero che $\kappa_i = \theta_j$, allora il suo tempo di rilevamento di equilibrio di Nash è pari al tempo/strategia ottimale $\rightarrow t_i^{ne} = \tilde{t}_j$. Se ciò non fosse così, allora t_i^{ne} è pari a 0 → significa che l'utente non partecipa all'attività MCS proposta.
- Dopo aver svolto tutte queste operazioni, il risultato atteso è un insieme di profili di strategie di Nash $t^{ne} = (t_1^{ne}, t_2^{ne}, \dots, t_n^{ne})$

Qui vi è la trascrizione/spiegazione:

1. Si ordinano i costi unitari degli utenti in un insieme, dove i costi sono disposti in ordine crescente;
2. Si costruisce un insieme inizialmente vuoto, che conterrà i costi unitari degli utenti che partecipano all'attività MCS;
3. Si assume che esista un certo indice che rappresenti il più piccolo valore tale che la somma delle frequenze associate ai costi unitari, da quell'indice in poi, sia maggiore di due. Questo implica che ci siano almeno due utenti idonei;
4. Aumentando progressivamente il valore dell'indice, si verifica per ciascun utente se la sua classe di costo unitario può far parte dell'insieme di partecipanti. Se il costo unitario di un utente risulta essere minore di un certo valore soglia (calcolato in base alla media pesata dei costi precedenti), allora si verifica se almeno un utente con quel costo soddisfa la condizione. In tal caso, si aggiunge l'utente all'insieme e si continua ad incrementare l'indice. Questo processo serve a selezionare gli utenti in base alla loro efficienza;
5. Si calcola quindi il numero totale di utenti selezionati (cioè quelli appartenenti al crowdsourcing), e per ciascun costo unitario si determina il tempo o la strategia ottimale di ciascun utente;
6. Si definisce poi un insieme di utenti idonei, che contiene gli indici degli utenti i cui costi unitari appartengono all'insieme precedentemente costruito;
7. Se un utente appartiene a questo insieme, e se il suo tempo di rilevamento all'equilibrio di Nash è uguale al tempo ottimale calcolato, allora partecipa. Se non è così, il tempo ottimale sarà pari a zero, il che significa che l'utente non partecipa all'attività MCS proposta;
8. Dopo aver svolto tutte queste operazioni, il risultato atteso è un insieme di profili di strategie che rappresentano l'equilibrio di Nash.

Algorithm 1: Computation of the NE

```

1 Sort the unit costs in  $\Theta$ ,  $\theta_1 < \theta_2 < \dots < \theta_l$ ;
2  $\Theta_w \leftarrow \emptyset$ ;
3 Let  $j \in [1, l]$  be the smallest such that  $\sum_{k=1}^j n_k \geq 2$ ;
4  $j \leftarrow j + 1$ ;
5 while  $j \leq l$  and  $\theta_j < \frac{\theta_j + \sum_{k=1}^{j-1} n_k \theta_k}{\sum_{k=1}^{j-1} n_k}$  do
6   if  $n_j > 0$  then  $\Theta_w \leftarrow \Theta_w \cup \{\theta_j\}$  ;
7    $j \leftarrow j + 1$ ;
8 end
9  $n_0 \leftarrow \sum_{\theta_k \in \Theta_w} n_k$ ;
10 foreach  $\theta_j \in \Theta_w$  do
11    $\tilde{t}_j = \frac{(n_0 - 1)R}{\sum_{\theta_k \in \Theta_w} n_k \theta_k} \left( 1 - \frac{(n_0 - 1) \theta_j}{\sum_{\theta_k \in \Theta_w} n_k \theta_k} \right)$ ;
12 end
13  $S \leftarrow \{i | \kappa_i \in \Theta_w\}$ ;
14 foreach  $i \in U$  do
15   if  $i \in S$  then  $t_i^{ne} = \tilde{t}_j$ , such that  $\kappa_i = \theta_j$  ;
16   else  $t_i^{ne} = 0$  ;
17 end
18 return  $t^{ne} = (t_1^{ne}, t_2^{ne}, \dots, t_n^{ne})$ 

```

Meccanismo di incentivazione per un modello utente-centrico

Il nostro primo tentativo consiste nel progettare un meccanismo di incentivazione che massimizzi l'utilità del crowdsourcer. In questo caso, progettare un meccanismo di incentivazione diventa un problema di ottimizzazione, chiamato **problema di selezione degli utenti**: dato un insieme \mathcal{U} di utenti, selezionare un sottoinsieme \mathcal{S} tale che $\widetilde{u}_0(\mathcal{S})$ sia massimizzato tra tutti i possibili sottoinsiemi. Inoltre, è chiaro che il pagamento di un utente deve coincidere con la sua offerta $p_i = b_i$ per massimizzare $\widetilde{u}_0(\mathcal{S})$. L'utilità \widetilde{u}_0 diventa quindi:

$$\widetilde{u}_0(\mathcal{S}) = v(\mathcal{S}) - \sum_{i \in \mathcal{S}} b_i$$

Per rendere il problema significativo, si assume che esista almeno un utente i tale che $\widetilde{u}_0(\{i\}) > 0$; tuttavia, è un problema NP difficile da affrontare al fine di trovare la soluzione ottimale al problema di selezione degli utenti, e quindi è necessario sviluppare algoritmi di approssimazione. A tal fine, sfruttiamo la **submodularità della funzione di utilità**.

Definizione 3 (Funzione submodulare): Sia \mathcal{X} un insieme finito. Una funzione $f: 2^{\mathcal{X}} \rightarrow \mathbb{R}$ è detta submodulare se $f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B)$ per ogni $A \subseteq B \subseteq \mathcal{X}$ e che $x \in \mathcal{X} \setminus B$.

Per dimostrare che la funzione di utilità del crowdsourcer \widetilde{u}_0 è **submodulare**, si parte dalla definizione: una funzione è submodulare se l'aggiunta di un elemento a un insieme piccolo porta un guadagno maggiore (o uguale) rispetto all'aggiunta dello stesso elemento a un insieme più grande. Nel nostro caso, vogliamo verificare che l'incremento dell'utilità \widetilde{u}_0 nel passare da un insieme \mathcal{S} a $\mathcal{S} \cup \{i\}$ sia minore o uguale all'incremento ottenuto passando da un insieme più grande T a $T \cup \{i\}$, dove $\mathcal{S} \subseteq T$ e l'utente i non è incluso in T . Poiché \widetilde{u}_0 è composta da due termini – uno positivo (beneficio v) e uno negativo (costo b) – e i costi sono uguali da entrambi i lati dell'equazione, possiamo concentrarci solo sul beneficio v . Analizzando il beneficio, si dimostra che l'aggiunta dell'utente i fornisce un guadagno minore se fatta su un insieme più grande, il che rispetta la proprietà della submodularità. Quindi, sia v che b sono funzioni submodulari.

Quando una funzione da massimizzare è **submodulare, monotona** (cioè non decresce) e **sempre positiva**, esistono algoritmi greedy (cioè che fanno scelte locali ottimali passo dopo passo) che garantiscono buoni risultati. Ad esempio, si può ottenere una soluzione che si avvicina a circa il 63% della migliore soluzione possibile, ovvero $1 - \frac{1}{e}$; tuttavia, nel nostro caso, l'utilità \widetilde{u}_0 può anche diventare negativa; quindi, non possiamo applicare direttamente questi algoritmi. Per risolvere il problema, si definisce una nuova funzione $f(\mathcal{S})$, che è semplicemente $\widetilde{u}_0(\mathcal{S})$ sommata a una costante (la somma di tutti i costi b_i degli utenti). Questa nuova funzione è sempre positiva e mantiene la submodularità; quindi, può essere usata al posto di \widetilde{u}_0 per trovare una soluzione.

Sulla base di queste considerazioni, viene progettato un meccanismo d'asta chiamato **LSB (Local Search-Based)**. Questo meccanismo funziona in modo **iterativo e locale**: cerca di migliorare la soluzione attuale aggiungendo un utente nuovo o rimuovendone uno esistente, se questo porta a un aumento della funzione $f(\mathcal{S})$.

È stato dimostrato che, per qualsiasi valore piccolo $\epsilon > 0$, questo algoritmo può trovare un sottoinsieme di utenti \mathcal{S} tale che la funzione $f(\mathcal{S})$ sia almeno pari a circa un terzo della funzione ottimale (cioè quella che si otterrebbe con la miglior combinazione possibile di utenti), con una piccola tolleranza in base alla dimensione n del problema.

Algorithm 2: LSB Auction

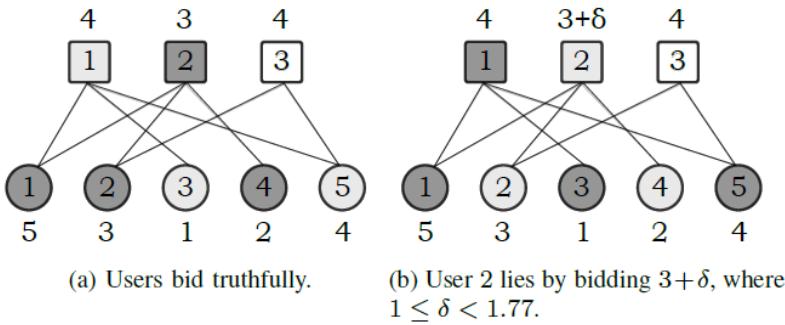
```

1  $\mathcal{S} \leftarrow \{i\}$ , where  $i \leftarrow \arg \max_{i \in \mathcal{U}} f(\{i\})$ ;
2 while there exists a user  $i \in \mathcal{U} \setminus \mathcal{S}$  such that
    $f(\mathcal{S} \cup \{i\}) > (1 + \frac{\epsilon}{n^2})f(\mathcal{S})$  do
3   |  $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$ ;
4 end
5 if there exists a user  $i \in \mathcal{S}$  such that
    $f(\mathcal{S} \setminus \{i\}) > (1 + \frac{\epsilon}{n^2})f(\mathcal{S})$  then
6   |  $\mathcal{S} \leftarrow \mathcal{S} \setminus \{i\}$ ; go to Line 2;
7 end
8 if  $f(\mathcal{U} \setminus \mathcal{S}) > f(\mathcal{S})$  then  $\mathcal{S} \leftarrow \mathcal{U} \setminus \mathcal{S}$  ;
9 foreach  $i \in \mathcal{U}$  do
10  | if  $i \in \mathcal{S}$  then  $p_i \leftarrow b_i$  ;
11  | else  $p_i \leftarrow 0$  ;
12 end
13 return  $(\mathcal{S}, p)$ 

```

Ora analizziamo il meccanismo d'asta **LSB (Local Search-Based)** utilizzando le **quattro proprietà desiderabili** come metriche di performance:

1. **Efficienza computazionale:** il tempo di esecuzione dell'algoritmo di ricerca locale LSB è $O\left(\frac{1}{\varepsilon} n^3 m \log m\right)$, dove la valutazione del valore della funzione f richiede un tempo pari a $O(m)$ e $|\mathcal{S}| \leq m$; pertanto, l'algoritmo LSB è computazionalmente efficiente.
2. **Razionalità individuale:** Il crowdsourcer paga ai vincitori l'importo che questi hanno offerto (b_i); quindi, l'asta LSB è **razionalmente individuale**, cioè i partecipanti non hanno una perdita netta.
3. **Redditività:** Dato che abbiamo assunto che esiste almeno un utente i tale che $\tilde{u}_0(\{i\}) > 0$ e che la funzione $f(\mathcal{S})$ **cresce strettamente** a ogni iterazione dell'algoritmo, possiamo garantire che $\tilde{u}_0(S) > 0$; quindi, l'algoritmo LSB è **redditizio** per il crowdsourcer.
4. **Veridicità:** Non sempre questo meccanismo conduce un utente a essere sincero, e ciò è dimostrabile sulla base di questo esempio:



Sia l'insieme degli utenti $U = \{1, 2, 3\}$, e l'insieme dei task $\Gamma = \{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5\}$. Le assegnazioni delle task agli utenti sono le seguenti:

- $\Gamma_1 = \{\tau_1, \tau_3, \tau_5\}$;
- $\Gamma_2 = \{\tau_1, \tau_2, \tau_4\}$
- $\Gamma_3 = \{\tau_2, \tau_5\}$

I costi dichiarati (bid) sono $c_1 = 4, c_2 = 3, c_3 = 4$ e che $\varepsilon = 0.1$.

Poniamo nel caso in cui tutti gli utenti sono sinceri. Avremo che:

$$f(\{1\}) = v(\Gamma_1) - b_1 + \sum_{i=1}^3 b_i = (5 + 1 + 4) - 4 + (4 + 3 + 4) = 17$$

$$f(\{2\}) = 18$$

$$f(\{3\}) = 14$$

Quindi, l'utente 2 è selezionato per primo.

$$f(\{2,1\}) = v(\Gamma_2 \cup \Gamma_1) - (b_2 + b_1) + \sum_{i=1}^3 b_i = 19$$

Poiché:

$$f(\{2,1\}) = 19 > \left(1 + \frac{0.1}{9}\right) \cdot f(\{2\}) = 18.2$$

anche l'utente 1 viene selezionato.

L'algoritmo si ferma qui perché non si può più migliorare f né aggiungendo un utente nuovo né rimuovendo un utente già selezionato. I pagamenti svolti sono stati i seguenti: $p_1 = b_1 = 4$ e $p_2 = b_2 = 3$

Adesso, supponiamo che l'utente 2 dichiari un'offerta falsata per trarne vantaggio: $b_2 = 3 + \delta$, con $1 \leq \delta < 1.77$. Allora avremo che:

$$f(\{1\}) = 17 + \delta, \quad f(\{2\}) = 18, \quad f(\{3\}) = 14 + \delta$$

Questa volta viene selezionato prima l'utente 1.

Poi:

$$f(\{1,2\}) = 19 > \left(1 + \frac{0.1}{9}\right) \cdot f(\{1\})$$

quindi viene selezionato anche l'utente 2.

L'algoritmo si ferma come prima, ma ora **l'utente 2 riceve un pagamento maggiore**, pari a $3 + \delta$ invece di 3.

Questo mostra che **mentire può portare a un guadagno maggiore**, quindi il meccanismo **non è veritiero**.

Sistema di asta IMCU

Anche se il meccanismo d'asta **LSB** è progettato per **massimizzare approssimativamente l'utilità del crowdsourcer**, la **mancanza di garanzia di veridicità** (in inglese *truthfulness*) lo rende meno attraente.

Poiché l'obiettivo finale è quello di progettare un meccanismo di incentivazione che **motivi gli utenti con smartphone a partecipare al crowdsensing**, impedendo però a chiunque di **manipolare il sistema truccando la propria offerta**, dobbiamo trovare un compromesso tra **massimizzazione dell'utilità e veridicità**.

Dunque, l'obiettivo è **progettare un meccanismo che soddisfi tutte e quattro le proprietà desiderabili, anche a costo di sacrificare parte dell'utilità del crowdsourcer**. Una possibile direzione consiste nel fare uso dei risultati già noti nel campo della **progettazione di meccanismi con vincoli di budget**. In questi meccanismi, il **pagamento ai vincitori** è trattato come **vincolo**, piuttosto che come parte della funzione obiettivo da massimizzare. Un'idea intuitiva potrebbe essere quella di provare **diversi valori di budget** all'interno del meccanismo e scegliere quello che fornisce la maggiore utilità; tuttavia, questa strategia potrebbe compromettere la **veridicità** del meccanismo di incentivazione.

Un meccanismo d'asta che soddisfa **tutte e quattro le proprietà desiderabili — efficienza computazionale, razionalità individuale, redditività e veridicità (truthfulness)** — può essere costruito **basandosi sul principio di caratterizzazione di Myerson**.

Cos'è la caratterizzazione di Myerson?

La **caratterizzazione di Myerson**, un risultato fondamentale nella **teoria dei meccanismi di asta (auction theory)**, fornisce **le condizioni necessarie e sufficienti** affinché un meccanismo d'asta sia **veritiero**. In parole semplici, stabilisce quando gli utenti **non hanno alcun incentivo a mentire sul proprio costo** e partecipano in modo trasparente e onesto.

Secondo la caratterizzazione di Myerson, un meccanismo d'asta è **veritiero (truthful)** se e solo se vengono rispettate **due condizioni fondamentali**:

1. La regola di selezione è monotona

Se un utente i vince l'asta offrendo un certo valore b_i , allora deve vincere anche **offrendo un valore inferiore** $b'_i \leq b_i$. Questo significa che **offrire un prezzo più basso non può penalizzare l'utente**, garantendo così trasparenza e correttezza.

2. Ogni vincitore riceve il proprio valore critico.

L'utente i deve essere **pagato un importo pari al valore massimo che avrebbe potuto offrire senza più vincere**. Dunque, se avesse fatto un'offerta **più alta rispetto a questo valore critico, non sarebbe stato selezionato**, e ciò assicura che il guadagno del partecipante non dipenda dalla manipolazione dell'offerta, ma dalla sua reale disponibilità.

Algorithm 3: IMCU Auction

```

1 // Phase 1: Winner selection
2  $\mathcal{S} \leftarrow \emptyset$ ,  $i \leftarrow \arg \max_{j \in \mathcal{U}} (v_j(\mathcal{S}) - b_j)$ ;
3 while  $b_i < v_i$  and  $\mathcal{S} \neq \mathcal{U}$  do
4   |  $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$ ;
5   |  $i \leftarrow \arg \max_{j \in \mathcal{U} \setminus \mathcal{S}} (v_j(\mathcal{S}) - b_j)$ ;
6 end
7 // Phase 2: Payment determination
8 foreach  $i \in \mathcal{U}$  do  $p_i \leftarrow 0$ ;
9 foreach  $i \in \mathcal{S}$  do
10  |  $\mathcal{U}_{-i} \leftarrow \mathcal{U} \setminus \{i\}$ ,  $\mathcal{T} \leftarrow \emptyset$ ;
11  | repeat
12  |   |  $i_j \leftarrow \arg \max_{j \in \mathcal{U}_{-i} \setminus \mathcal{T}} (v_j(\mathcal{T}) - b_j)$ ;
13  |   |  $p_i \leftarrow \max\{p_i, \min\{v_i(\mathcal{T}) - (v_{i_j}(\mathcal{T}) - b_{i_j}), v_i(\mathcal{T})\}\}$ ;
14  |   |  $\mathcal{T} \leftarrow \mathcal{T} \cup \{i_j\}$ ;
15  | until  $b_{i_j} \geq v_{i_j}$  or  $\mathcal{T} = \mathcal{U}_{-i}$ ;
16  | if  $\mathcal{T} = \mathcal{U}_{-i}$  then  $p_i \leftarrow \max\{p_i, v_i(\mathcal{T})\}$ ;
17 end
18 return  $(\mathcal{S}, p)$ 

```

L'algoritmo IMCU è stato progettato per soddisfare tutte e quattro le proprietà desiderabili di un buon meccanismo d'asta per il crowdsensing con smartphone:

- **Efficienza computazionale**
- **Razionalità individuale**
- **Redditività**
- **Veridicità (truthfulness)**

Per raggiungere questi obiettivi, l'algoritmo è diviso in due fasi principali:

1. **Selezione dei vincitori**
2. **Determinazione dei pagamenti**

Fase 1: Selezione dei vincitori (Winner Selection)

Questa fase utilizza un approccio **greedy**: vengono selezionati, uno alla volta, gli utenti che forniscono il massimo incremento marginale al valore della funzione obiettivo, tenendo conto del costo (cioè dell'offerta) che presentano.

Passaggi:

1. Si inizializza l'insieme dei vincitori S come vuoto.
2. A ogni iterazione si sceglie l'utente i che massimizza la differenza tra il proprio valore marginale $v_i(S)$ e la sua offerta b_i .
3. Se la differenza è positiva $v_i(S) > b_i$ e non sono stati ancora considerati tutti gli utenti, l'utente i viene aggiunto a S .
4. Il processo si ripete finché nessun nuovo utente può aumentare il valore della funzione obiettivo con una differenza positiva.

Definizione di valore marginale:

Per un utente i , il valore marginale è:

$$v_i(S) = v(S \cup \{i\}) - v(S)$$

Ovvero, quanto contribuisce in più i all'insieme S già esistente. Grazie alla **submodularità** della funzione di valore (cioè: rendimenti marginali decrescenti), gli utenti sono implicitamente ordinati in modo che:

$$c_1 - b_1 \geq c_2 - b_2 \geq \dots \geq c_n - b_n$$

L'insieme dei vincitori finali sarà quindi $S = \{1, 2, \dots, L\}$, dove L è il massimo indice per cui $v_L - b_L > 0$.

Fase 2: Determinazione dei pagamenti (Payment Computation)

Una volta selezionati i vincitori, per ciascuno si calcola il **prezzo critico**: l'offerta massima che l'utente avrebbe potuto fare, restando comunque selezionato. Questo garantisce la **veridicità** del meccanismo.

Per ogni vincitore i :

1. Si considera l'insieme degli altri utenti $U - \{i\}$.
2. Si itera su questo insieme, costruendo progressivamente un sottoinsieme T , e a ogni passo si seleziona l'utente j che massimizza $v_j(T) - b_j$.
3. In parallelo, si calcola il massimo prezzo che l'utente i avrebbe potuto offrire per essere selezionato al posto di j .
4. Si ripete fino a che $b_j \geq v_j(T)$ oppure $T = U - \{i\}$.
5. Se T include tutti gli utenti, si assegna al vincitore i il valore massimo $v_i(T)$.

In questo modo, si costruisce p_i come il massimo valore tra tutte le offerte critiche calcolate durante i passaggi.

Le proprietà dimostrate dell'algoritmo IMCU sono le seguenti:

1. **Efficienza Computazionale:** Il tempo richiesto è $O(nm^3)$, dove n è il numero di utenti e m il numero di task. Le operazioni più costose sono nel ciclo che calcola i pagamenti, che viene eseguito per ciascun vincitore. È un limite superiore conservativo: nella pratica, $m \ll n$, quindi il tempo di esecuzione è spesso molto più basso.
2. **Razionalità Individuale:** ogni vincitore riceve un pagamento **maggiore o uguale** alla propria offerta, e ciò è garantito perché il pagamento è calcolato come il massimo tra varie offerte critiche, mai inferiori a bi.
3. **Redditività:** Il meccanismo garantisce che la **somma dei valori ottenuti sia superiore alla somma dei pagamenti**: $\sum v_i > \sum p_i$. Questo si può dimostrare facendo in modo che per ogni utente il pagamento non supera il valore marginale che l'utente apporta.
4. **Veridicità (Truthfulness):** Il meccanismo è veritiero secondo la caratterizzazione di **Myerson**, perché:
 - a. La regola di selezione è **monotona**: offrire meno non penalizza l'utente.
 - b. Il pagamento è il **valore critico**: l'offerta massima che l'utente avrebbe potuto fare per restare vincitore.

Struttura della Simulazione e Analisi dei Risultati

Al fine di valutare l'efficacia dei meccanismi di incentivazione proposti, sono stati definiti due modelli distinti: uno centrato sul crowdsourcer e uno centrato sull'utente, entrambi implementati in ambienti simulati con parametri controllati.

Nel **modello centrato sul crowdsourcer (IMCC)**, si è ipotizzato che il costo di ciascun utente fosse distribuito uniformemente nell'intervallo $[1, \tau_{\max}]$, con τ_{\max} variabile da 1 a 10 (incremento di 1). Il numero di utenti n è stato fatto variare da 100 a 1000 con incremento di 100. La funzione di utilità del crowdsourcer è stata definita come:

$$g(\tilde{t}_1, \tilde{t}_2, \dots, \tilde{t}_l; n_1, n_2, \dots, n_l) = \lambda \cdot \log \left(1 + \sum_{\theta_j \in \Theta} n_j \cdot \log(1 + \tilde{t}_j) \right)$$

dove $\lambda = 10$ è un parametro di sistema, \tilde{t}_j rappresenta l'output lavorativo di un utente con costo unitario c_j , e il logaritmo esterno riflette i rendimenti decrescenti rispetto al numero di utenti partecipanti.

Per il **modello centrato sull'utente (IMCU)**, è stata considerata un'applicazione di copertura del segnale cellulare, dove gli utenti misurano il segnale in punti geolocalizzati. Sono stati utilizzati due dataset:

- *Random*: utenti e task distribuiti casualmente in un'area di $1000m \times 1000m$; ogni utente esegue i task presenti entro 30m da sé. Il numero degli utenti n varia da 1000 a 10000, mentre i task m da 100 a 500.
- *Manhattan*: area reale di Manhattan (NY), con 389 POI (ristoranti, scuole, biblioteche ecc.) raccolti da Google Maps. Gli utenti si muovono secondo un modello simile al "*Manhattan mobility model*", rilevando il segnale ogni 6 metri. I compiti sono assegnati se entro 30m dal punto di rilevamento. In questo caso, il numero di utenti n varia da 200 a 1000.

Il valore di ciascuna task è distribuito uniformemente tra 1 e 5. I costi degli utenti sono determinati come

$$c_i = \rho |\Gamma_i|$$

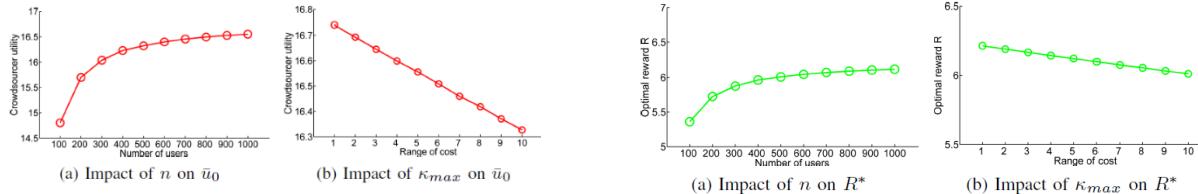
dove ρ è distribuito uniformemente tra 1 e 10. È stato fissato $\varepsilon = 0.01$ per LSB. Le distribuzioni possono essere adattate anche a forme diverse (es. gaussiane) senza modificare significativamente i risultati.

Risultati Osservati

Durante le simulazioni con il modello IMCC, è stato notato che il numero di utenti partecipanti tende a diminuire quando i costi unitari sono più eterogenei. Questo perché, secondo la condizione implementata nel ciclo while, se tutti gli utenti hanno costi simili, tendono tutti a partecipare; al contrario, una maggiore variabilità nei costi comporta l'esclusione degli utenti con costi più elevati, che non soddisfano più la condizione.

Per quanto riguarda l'utilità del crowdsourcer, i risultati mostrano un chiaro comportamento di rendimenti decrescenti:

- Quando si aumenta il numero di utenti (n), l'utilità inizialmente cresce, ma con una progressiva diminuzione del tasso di incremento;
- Quando aumenta la variabilità dei costi (τ_{\max}), l'utilità complessiva del crowdsourcer tende a diminuire.



Relativamente alla ricompensa ottimale (R^*), è stato osservato che:

- R^* cresce con n , ma raggiunge un punto in cui si stabilizza;
- All'aumentare della diversità nei costi, la ricompensa ottimale per incentivare gli utenti diminuisce, in quanto meno utenti sono in grado di partecipare.

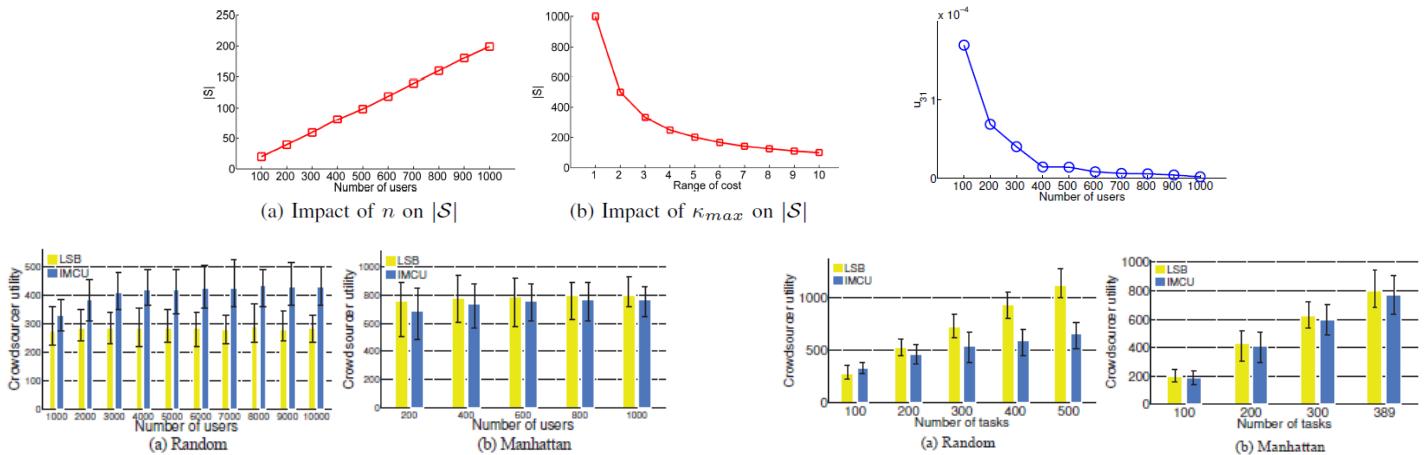


Fig. 10. Impact of n on crowdsourcer utility

Fig. 11. Impact of m on crowdsourcer utility

Nel caso del modello centrato sull'utente (IMCU), è stato condotto un confronto diretto con LSB. I risultati mostrano che:

- IMCU garantisce utilità più alta per il crowdsourcer quando il numero di task m è relativamente piccolo rispetto al numero di utenti n ;
- Quando m cresce e diventa comparabile o superiore a n , la performance di IMCU cala, e il divario con LSB aumenta;
- Anche in questo modello, l'utilità del crowdsourcer segue un comportamento a rendimenti decrescenti all'aumentare di n .