From Social Trust Assisted Reciprocity (STAR) to Utility-Optimal Mobile Crowdsensing

Xiaowen Gong*, Xu Chen*, Junshan Zhang*, H. Vincent Poor†

*School of Electrical, Computer and Energy Engineering

Arizona State University, Tempe, AZ 85287

Email: {xgong9, xchen179, junshan.zhang}@asu.edu

†Department of Electrical Engineering

Princeton University, Princeton, NJ 08544

Email: poor@princeton.edu

***Abstract*—Crowdsensing has been widely recognized as a promising paradigm for numerous applications in mobile networks. To realize the full benefit of crowdsensing, one fundamental challenge is to incentivize users to participate. In this paper, we leverage social trust assisted reciprocity (STAR), a synergistic marriage of social trust and reciprocity, to develop an incentive mechanism in order to stimulate users' participation. We investigate thoroughly the efficacy of STAR for satisfying users' sensing requests, for a given social tie structure among users. Specifically, we first show that all requests can be satisfied if and only if sufficient social credit can be transferred from users who request more sensing services than what they can provide to users who can provide more than what they request. Then we investigate utility maximization for sensing services, and show that it boils down to maximizing the utility of a circulation flow in the combined graph of the social graph and request graph. Accordingly, we develop an algorithm that iteratively cancels cycles of positive weights in the residual graph, and thereby finds the optimal solution efficiently.**

I. Introduction

Mobile crowdsensing has recently emerged as a promising paradigm for a variety of applications, thanks to the pervasive penetration of mobile devices in people's lives [1]. Although the benefit of crowdsensing is pronounced, a user would not participate in sensing without receiving adequate incentive. Therefore, effective *incentive design* is essential for realizing the benefit of crowdsensing. Recent studies on incentive design for crowdsensing (e.g., [2]–[4]) mostly use monetary reward to stimulate users' participation, which rely on a global (virtual) currency system that typically incurs significant implementation overhead. Therefore, it is appealing to design a crowdsensing system that can motivate users to participate without using a global currency, which is a subject of this study.

Mobile users' behaviors are increasingly influenced by their social relationships, mainly due to the rapid growth of online social networking services [5]. As an important aspect of social relationship, *social trust* can be exploited to stimulate crowdsensing: if Alice has social trust in Bob, then Alice is willing to help Bob since Alice can trust Bob, in the belief that Bob would help Alice in the future to return the favor.

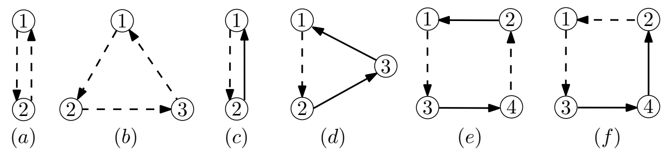

Fig. 1. Examples of social trust assisted reciprocity cycles. (a)-(d) are special cases: (a) direct reciprocity cycle; (b) indirect reciprocity cycle; (c) direct social trust based cycle; (d) indirect social trust based cycle. Solid edges are social edges. Dashed edges are request edges.

In this paper, we devise an incentive mechanism to stimulate users' participation in crowdsensing, by using *Social Trust Assisted Reciprocity* (STAR) - a synergistic marriage of social trust and reciprocity. The basic idea of STAR is that Alice is willing to help Bob if someone who trusts Bob can help someone trusted by Alice (as illustrated in Fig. 1). This is because that the overhead of Alice for helping Bob is compensated, as the one trusted by Alice will help Alice in the future to return the favor. By taking advantage of reciprocity ("synchronous exchange") with the assist of social trust ("asynchronous exchange"), STAR can efficiently encourage users' participation in crowdsensing. Furthermore, compared to traditional currency-based schemes, STAR can incur a much lower implementation overhead due to the use of the existing social trust.

The main thrust of this study is devoted to characterizing the fundamental performance of STAR, particularly for satisfying users' sensing requests given the social trust structure among them. We are interested in answering two important questions: *What are the conditions under which all requests can be satisfied? What is the maximum utility that can be achieved by provided service?* These two questions are similar in spirit to admission control and network utility maximization, respectively.

We summarize the main contributions of this paper as follows.

- We design STAR, an incentive mechanism which stimulates users' participation by using *social trust assisted reciprocity* (STAR). We investigate thoroughly the efficacy of STAR for satisfying users' sensing requests, for a given social trust structure among users. Specifically, we first show that all requests can be satisfied if and only if users who request more sensing service than what they can provide can transfer sufficient social credit to those users who can provide more than what they request. Then we investigate utility maximization for sensing service, and show that this problem is equivalent to maximizing the utility of a *circulation flow* in the combined social graph and request graph. Based on this observation, we develop an algorithm that iteratively cancels the cycles of positive weights in the *residual graph*, and thereby finds the optimal solution efficiently.

The rest of this paper is organized as follows. In Section II, we design an incentive mechanism based on social trust assisted reciprocity (STAR). Based on STAR, Section III investigates the conditions for satisfying all sensing requests and the utility maximization for sensing service. The paper is concluded in Section IV.

II. STAR: Social Trust Assisted Reciprocity Based Incentive Mechanism

A. System Model

We consider a crowdsensing system consisting of a set of mobile users $V = \{1, \cdots, N\}$. Each user can request a certain amount of sensing service from another user. We model users' sensing requests by a *request graph* $G^R \triangleq (V, E^R)$, where user i and user j are connected by a directed *request edge* $e_{ij}^R \in E^R$ if user j requests sensing service[1] from user i. The capacity $R_{ij} > 0$ of each request edge e_{ij}^R represents the *amount* of service requested by user j from user i. The flow $f_{ij}^R > 0$ on the request edge e_{ij}^R represents the amount of service provided by user i to user j.

A user obtains utility from its requested sensing service, which depends on the amount of service it receives from each user requested by her. We assume that user j obtains a utility of U_{ij} for each *unit* amount of service provided by user i. We further assume that a user's utility is equal to the total utility of the service provided to her. More complex forms of utility will be studied in future work.

We model the social trust structure among users by a *social graph* $G^S \triangleq (V, E^S)$, where user i and user j are connected by a directed *social edge* $e_{ij}^S \in E^S$ if user j has social trust in user i. The capacity $S_{ij} > 0$ of each social edge e_{ij}^S represents the *social credit limit*, which specifies the maximum amount of social credit that can be transferred from user i to user j. The flow f_{ij}^S on the social edge e_{ij}^S represents the amount of social credit transferred between user i and user j. Note that $f_{ij}^S = -f_{ji}^S$ holds for each pair of social edges between two users, where $f_{ij}^S > 0$ or $f_{ji}^S > 0$ indicates that credit of amount f_{ij}^S or f_{ji}^S is transferred from user i to user j or from user j to user i, respectively.

[1]For brevity, we use "sensing service" and "service" interchangeably throughout the paper.

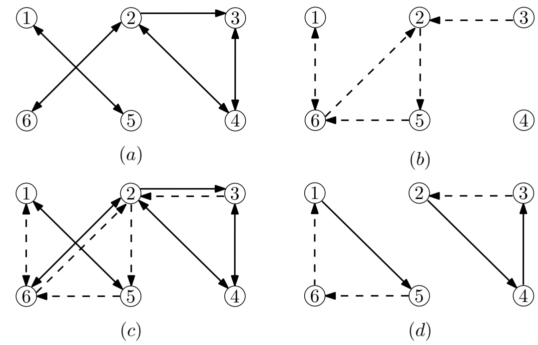

Fig. 2. An example of (a) the social graph; (b) the request graph; (c) the combined social and request graph; (d) two STAR cycles in the social-request graph.

B. Design Description

The basic incentive structure of the STAR mechanism is a *social trust assisted reciprocity cycle* (STAR) in which a set of users have incentive to provide service. It is defined in the combined social and request (social-request) graph $G \triangleq (V, E^S \cup E^R)$ (as illustrated in Fig. 2).

Definition 1: A social trust assisted reciprocity cycle is a directed cycle in the social-request graph G.

In a STAR cycle, a user is willing to provide service since *the overhead is compensated by receiving credit or service from another user in the cycle*. For each user in a STAR cycle, the amount of service or credit it receives should be *equal* to that of service or credit it provides or spends, respectively. Let f_c denote a *balanced* flow along a STAR cycle c, which has the same flow value on each edge in c. The flow on a social or request edge in the *aggregate* flow f of a set of balanced flows $\{f_c, c \in \mathcal{C}\}$ along cycles $\mathcal{C}$ is given by

$$f_{ij}^S = \sum_{c \in \mathcal{C}: e_{ij}^S \in c} f_c - \sum_{c \in \mathcal{C}: e_{ji}^S \in c} f_c, \quad f_{ij}^R = \sum_{c \in \mathcal{C}: e_{ij}^R \in c} f_c$$

respectively. Note that the credit transferred from user i to j (i.e., the flow on $e_{ij}^S \in E^S$) in the balanced flow along a STAR cycle can be partly or completely *canceled* by that from user j to i in another STAR cycle. Users can participate in a set of balanced flows along STAR cycles if and only if the aggregate flow satisfies the capacity constraints on request and social edges.

Definition 2: A set of balanced flows along STAR cycles is feasible if the aggregate flow satisfies the following capacity constraints:

$$-S_{ji} \le f_{ij}^S \le S_{ij}, \; f_{ji}^S = -f_{ij}^S, \; \forall e_{ij} \in E^S \quad (1)$$

$$0 \le f_{ij}^R \le R_{ij}, \; \forall e_{ij} \in E^R. \quad (2)$$

Under the STAR mechanism, all users are willing to participate in any feasible set of balanced flows along STAR cycles.

III. Exploiting STAR to Satisfy Sensing Requests

A. Satisfying All Sensing Requests

Based on STAR cycles, we first show that it suffices to focus on *circulation* flows in the social-request graph

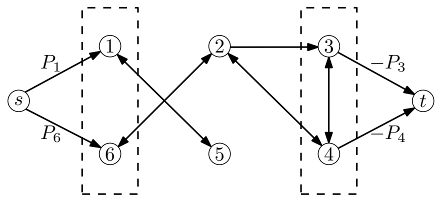

Fig. 3. The extended social graph constructed from the social graph in Fig. 2 where $P_1 > 0$, $P_2 = 0$, $P_3 < 0$, $P_4 < 0$, $P_5 = 0$, $P_6 > 0$. The notation next to an edge is its capacity.

defined as follows.

Definition 3: A flow f in the social-request graph G is a circulation if f satisfies the capacity constraints (1), (2), and the flow conservation constraints

$$\sum_{j:e_{ij}\in E^R} f_{ij}^R + \sum_{j:e_{ij}\in E^S} f_{ij}^S = \sum_{j:e_{ji}\in E^R} f_{ji}^R,\ \forall i \in V. \qquad (3)$$

It is clear that the aggregate flow of any feasible set of balanced flows along STAR cycles is a circulation flow in G. The following lemma shows that the converse is also true.

Lemma 1: Any circulation flow in the social-request graph amounts to the aggregate flow of a feasible set of balanced flows along STAR cycles.

Due to space limitation, all the proofs of this paper are given in our online technical report [6]. We define P_i as the total amount of service requested by user i minus the amount that user i can provide:

$$P_i \triangleq \sum_{j:e_{ji}\in E^R} R_{ji} - \sum_{j:e_{ij}\in E^R} R_{ij}.$$

Then we construct an extended social graph G^{S^+} from the social graph G^S by adding a directed edge with capacity P_i from a virtual source node s to each node i with $P_i > 0$, and adding a directed edge with capacity $-P_i$ from each node i with $P_i < 0$ to a virtual destination node t (as illustrated in Fig. 3). Let P be defined as

$$P \triangleq \sum_{i:P_i>0} P_i = -\sum_{i:P_i<0} P_i.$$

Theorem 1: All sensing requests can be satisfied under STAR if and only if P is equal to the maximum flow value from s to t in the extended social graph G^{S^+}.

Remark 1: Theorem 1 provides a useful insight: all requests can be satisfied if and only if *users who request more service than what they can provide can transfer sufficient social credit to users who can provide more than what they request, to compensate their imbalance in requests*. Intuitively speaking, the social graph serves as a "buffer" to partially or completely "absorb" the mismatch among users' requests. It is worth noting that the maximum amount of service provided under STAR is in general *not* equal to the maximum flow value from s to t in G^{S^+}.

Remark 2: We note that an important difference between [7] and our study is that the results in [7] is based on the assumption that *all users are connected in the social network*, whereas our model here does not have this assumption. This is essentially because that reciprocity is used in STAR but not in [7].

B. Utility Maximization for Sensing Service

Due to the mismatch of sensing service requests and social credit limits, it is possible that not all requests can be satisfied. In this case, a natural objective is to maximize the total utility of provided service. The next result follows from Lemma 1.

Theorem 2: The maximum utility of sensing service provided under STAR is equal to the maximum utility of a circulation flow in the social-request graph.

Note that the flow on a social edge does not generate any utility. By Theorem 2, our problem can be written as

$$\underset{f_{ij}^S, f_{ij}^R}{\text{maximize}} \quad \sum_{i,j:e_{ij}\in E^R} U_{ij} f_{ij}^R \qquad (4)$$

$$\text{subject to} \quad \text{constraints } (1), (2), (3).$$

Note that we can *maximize the total amount of service provided under STAR by solving problem* (4) *with the utility* U_{ij} *set to 1 for each request edge* e_{ij}^R.

In the following, we will solve problem (4) using an algorithm inspired by the *cycle-canceling* algorithm for solving the minimum cost flow problem [8]. We should note that problem (4) is very different from a typical network flow problem in that two nodes can be connected by *multiple* edges (request edges and social edges). Furthermore, *request edges and social edges carry different types of flows*: the flows on all request edges are *non-negative* and *independent* (as in constraint (2)), while the flows on social edges can be *negative* and must be *inverse* between a pair of users (as in constraint (1)).

We start with constructing a *residual graph* $G_f \triangleq (V, E_f^S \cup E_f^R)$ of the social-request graph G for a given flow f. Specifically, for each request edge $e_{ij}^R \in E^R$, we construct a *forward* edge $\overrightarrow{e}_{ij}^R \in E_f^R$ and a *backward* edge $\overleftarrow{e}_{ji}^R \in E_f^R$ with capacity

$$\overrightarrow{R}_{ij} = R_{ij} - f_{ij}^R,\ \overleftarrow{R}_{ij} = f_{ij}^R$$

respectively. For each *pair* of social edges $e_{ij}^S, e_{ji}^S \in E^S$, we construct a pair of edges $\overrightarrow{e}_{ij}^S, \overrightarrow{e}_{ji}^S \in E_f^S$ with capacity

$$\overrightarrow{S}_{ij} = S_{ij} - f_{ij}^S,\ \overrightarrow{S}_{ji} = S_{ji} - f_{ji}^S$$

respectively. We do *not* construct an edge in the residual graph if its capacity is zero. Then we set the *weight* of each forward edge $\overrightarrow{e}_{ij}^R \in E_f^R$ and each backward edge $\overleftarrow{e}_{ji}^R \in E_f^R$ as

$$\overrightarrow{W}_{ij}^R = U_{ij},\ \overleftarrow{W}_{ij}^R = -U_{ij}$$

respectively. The weights of each pair of edges $\overrightarrow{e}_{ij}^S, \overrightarrow{e}_{ji}^S \in E_f^S$ are set to

$$\overrightarrow{W}_{ij}^S = \overrightarrow{W}_{ji}^S = 0.$$

The following lemma establishes the optimality condition for solving problem (4).

Lemma 2: A flow f is optimal for problem (4) if and only if there exists no cycle of positive weight in the residual graph G_f.

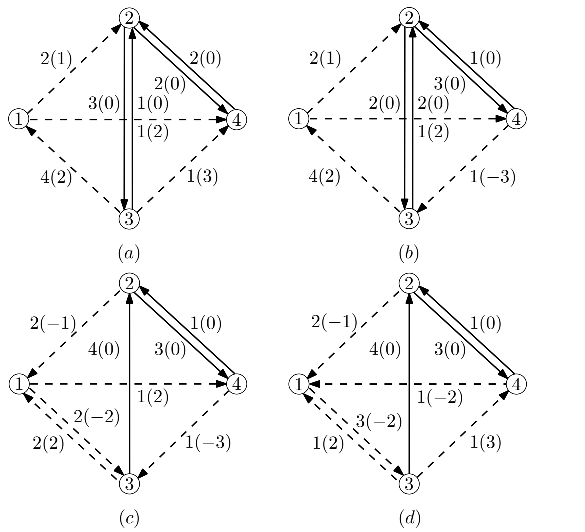

Fig. 4. An example of running Algorithm 1. (a) Initial social-request graph with the empty flow; (b) Residual graph after augmenting with a flow of value 1 along cycle $2 \to 3 \to 4 \to 2$; (c) Residual graph after augmenting with a flow of value 2 along cycle $1 \to 2 \to 3 \to 1$; (d) Residual graph after augmenting with a flow of value 1 along cycle $1 \to 4 \to 3 \to 1$. For each edge, the number before () is its capacity; the number in () is its weight.

Using Lemma 2, we can develop an algorithm as described in Algorithm 1 to solve problem (4). The algorithm starts with the empty flow in the network. It iteratively searches for a cycle of *positive weight* in the residual graph and cancels the cycle by augmenting the current flow in the graph with a balanced flow along the cycle, until no cycle of positive weight exists. In each iteration, the value of the flow to augment with is set to be the *residual capacity* of the cycle, which is the minimum capacity of all edges in that cycle. We show how Algorithm 1 works by an illustrative example in Fig. 4.

As for step 2 in Algorithm 1, we can use an algorithm similar to the Bellman-Ford algorithm [9] to find a cycle of positive weight in the residual graph, if there exists one. For ease of exposition, we will focus on problem (4) with *rational* parameters: the utilities and capacities of all social and request edges are rational numbers. This setting is of important interest in general, since the parameters of most practical problems are rational numbers. Then problem (4) with rational parameters can be equivalently converted to one with *integral* parameters by multiplying with a suitably large integer K. The solution of the original problem (with rational parameters) is equal to the solution of the new problem (with integral parameters) divided by K.

For problem (4) with rational parameters, let $\overline{U}$ and $\overline{R}$ denote the maximum utility and maximum capacity of a request edge, respectively (i.e., $\overline{U} = \max_{e_{ij} \in E^R} U_{ij}$, $\overline{R} = \max_{e_{ij} \in E^R} R_{ij}$). The following theorem shows that Algorithm 1 is correct and efficient.

Theorem 3: For problem (4) with divisible sensing service and rational parameters, Algorithm 1 finds the optimal flow and has running time $O(|V||E^R|(|E^R| + |E^S|)\overline{RU}K^2)$.

Algorithm 1: Find the optimal flow for problem (4) in social-request graph G

input : Social-request graph G
output: The optimal flow for problem (4)
1 Initialize an empty flow f in G;
2 **while** *There exists a cycle of positive weight in the residual graph G_f of flow f* **do**
3   Find a cycle c of positive weight in G_f;
4   Compute the residual capacity r_c of cycle c;
5   Augment flow f with a balanced flow of value r_c along cycle c;
6 **end**
7 **return** Flow f;

Remark 3: It is worth noting that, when sensing service is *indivisible* such that its amount has to be an integer, problem (4) is essentially an integer linear program, which is NP-hard to solve in general. In this case, we can still use Algorithm 1 to find the optimal flow for problem (4). In other words, using a network flow approach, we can capture and exploit the specific combinatorial structure of problem (4), based on which a polynomial-time algorithm can be developed to solve it.

IV. Conclusion

In this paper, we have designed STAR, an incentive mechanism using a synergistic marriage of social trust and reciprocity. Based on the STAR mechanism, we have established the conditions under which all sensing requests can be satisfied. We have also developed an efficient algorithm to maximize the utility of sensing service provided under STAR. An interesting research direction underway is to leverage the social tie structure among mobile users to stimulate their altruistic behaviors for cooperative networking.

References

[1] R. K. Ganti, Y. Fan, and L. Hui, “Mobile crowdsensing: current state and future challenges,” *IEEE Communications Magazine*, pp. 32–39, 2011.

[2] D. Yang, G. Xue, X. Fang, and J. Tang, “Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing,” in *ACM MOBICOM 2012*.

[3] T. Luo and C.-K. Tham, “Fairness and social welfare in incentivizing participatory sensing,” in *IEEE SECON 2012*.

[4] I. Koutsopoulos, “Optimal incentive-driven design of participatory sensing systems,” in *IEEE INFOCOM 2013*.

[5] “eMarketer: Social networking reaches nearly one in four around the world.” [Online]. Available: http://www.emarketer.com/Article/Social-Networking-Reaches-Nearly-One-Four-Around-World/1009976

[6] X. Gong, X. Chen, J. Zhang, and H. V. Poor, “From social trust assisted reciprocity (STAR) to utility-optimal crowdsensing in mobile networks,” Technical Report. [Online]. Available: http://informationnet.asu.edu/pub/STAR-globalsip14-TR.pdf

[7] Z. Liu, H. Hu, Y. Liu, K. W. Ross, Y. Wang, and M. Mobius, “P2P trading in social networks: The value of staying connected,” in *IEEE INFOCOM 2010*.

[8] R. Ahuja, T. Magnanti, and J. Orlin, *Network flows: Theory, algorithms, and applications*. Prentice Hall, 1993.

[9] X. Huang, “Negative-weight cycle algorithms,” in *2006 International Conference on Foundations of Computer Science*.

APPENDIX

Proof of Lemma 1

Consider a non-empty circulation flow f. We can find a node v_1 with a positive flow on an outgoing edge from v_1 and trace along this edge to another node v_2. Due to the flow conservation constraint, we can find an outgoing edge from v_2 with a positive flow and trace along it to a node v_3. We continue this tracing process until we visit a node v_j that has been visited before, i.e., $v_i = v_j$ for some $i < j$, and hence we find a STAR cycle $v_i \to v_{i+1} \to \cdots \to v_j$. Then we subtract flow f by a balanced flow along this cycle with value equal to the minimum flow value on an edge in that cycle. Thus the remaining flow is still a circulation flow in which the number of edges with non-zero flows is reduced. We can repeat this argument to subtract the remaining flow by a balanced flow along a cycle until it is empty. This implies that flow f is the aggregate flow of the subtracted balanced flows along the cycles, which is also feasible.

Proof of Theorem 1

By Lemma 1, all requests can be satisfied if and only if there is a circulation flow f in the social-request graph G that saturates all request edges (i.e., $f_{ij}^R = R_{ij}, \forall e_{ij}^R \in E^R$).

We first show the "if" part. Suppose S is equal to the value of the maximum flow f^{S^+} from s to t in G^{S^+}. Let f^S be the flow comprised of the flows on the social edges E^S in f^{S^+} (i.e., not including the edges from s and to t in G^{S^+}). Let f^R be the flow in the request graph G^R that saturates all request edges. Then we augment flow f^S in the social-request graph G with flow f^R to obtain a flow f in G. According to the construction of G^{S^+}, we have $\sum_{j:e_{ij}^S \in E^S} f_{ij}^S = P_i$ for each node $i \in V$, while we also have $\sum_{j:e_{ji}^R \in E^R} f_{ji}^R - \sum_{j:e_{ij}^R \in E^R} f_{ij}^R = P_i$. This shows that f is a circulation flow.

Next we show the "only if" part. Suppose f is a circulation flow in G that saturates all request edges. Let f^S be the flow comprised of the flows on the social edges E^S in f. Then we augment flow f^S with saturated flows on the edges from s and to t in G^{S^+} to obtain a flow f^{S^+} in G^{S^+}. According to the construction of G^{S^+}, f^{S^+} is a flow in G^{S^+} satisfying the capacity and flow conservation constraints, with a flow value of P from s to t.

Proof of Lemma 2

The "only if" part is easy to show: If there exists a cycle of positive weight in G_f, then we can augment the flow f with a balanced flow of value $\epsilon > 0$ along that cycle to construct a circulation flow with larger utility.

Next we show the "if" part. Suppose there exists no cycle of positive weight in G_f but there exists a circulation flow f' in G with larger utility than f. Similar to the residual graph G_f, we construct a graph $\overline{G} \triangleq (V, \overline{E}^S \cup \overline{E}^R)$ from G by constructing $\overrightarrow{e}_{ij}^R, \overleftarrow{e}_{ij}^R \in \overline{E}^R$ for each $e_{ij}^R \in E^R$ and $\overrightarrow{e}_{ij}^S, \overrightarrow{e}_{ji}^S \in \overline{E}^S$ for each pair of $e_{ij}^S, e_{ji}^S \in E^S$, and setting their weights the same as those in G_f. The difference between G_f and $\overline{G}$ is that all the edges are constructed in $\overline{G}$ (an edge is not constructed in G_f if its capacity is 0) and have unlimited capacities. Therefore, the edges in G_f is a subset of the edges in $\overline{G}$. Then we can define a flow g in $\overline{G}$ by defining the flows in g on the edges of $\overline{G}$ as

$$\overrightarrow{g}_{ij}^R = \max\{0, f_{ij}'^R - f_{ij}^R\}, \ \forall \overrightarrow{e}_{ij}^R \in \overline{E}^R$$
$$\overleftarrow{g}_{ij}^R = \max\{0, f_{ij}^R - f_{ij}'^R\}, \ \forall \overleftarrow{e}_{ij}^R \in \overline{E}^R$$
$$\overrightarrow{g}_{ij}^S = f_{ij}'^S - f_{ij}^S, \ \forall \overrightarrow{e}_{ij}^S \in \overline{E}^S.$$

It follows from the definition that

$$\overrightarrow{g}_{ij}^R - \overleftarrow{g}_{ij}^R = f_{ij}'^R - f_{ij}^R, \ \forall e_{ij}^R \in E^R.$$

Then the net flow value at each node $i \in V$ in flow g is

$$\sum_{j:\overrightarrow{e}_{ij}^R \in \overline{E}^R} \overrightarrow{g}_{ij}^R + \sum_{j:\overleftarrow{e}_{ji}^R \in \overline{E}^R} \overleftarrow{g}_{ji}^R + \sum_{j:\overrightarrow{e}_{ij}^S \in \overline{E}^R} \overrightarrow{g}_{ij}^S - \sum_{j:\overleftarrow{e}_{ij}^R \in \overline{E}^R} \overleftarrow{g}_{ij}^R - \sum_{j:\overrightarrow{e}_{ji}^R \in \overline{E}^R} \overrightarrow{g}_{ji}^R$$
$$= \sum_{j:e_{ij}^R \in E^R} \left(f_{ij}'^R - f_{ij}^R\right) - \sum_{j:e_{ji}^R \in E^R} \left(f_{ji}'^R - f_{ji}^R\right) + \sum_{j:e_{ij}^S \in E^S} \left(f_{ij}'^S - f_{ij}^S\right)$$
$$= \left(\sum_{j:e_{ij}^R \in E^R} f_{ij}'^R + \sum_{j:e_{ij}^S \in E^S} f_{ij}'^S - \sum_{j:e_{ji}^R \in E^R} f_{ji}'^R \right)$$
$$- \left(\sum_{j:e_{ij}^R \in E^R} f_{ij}^R + \sum_{j:e_{ij}^S \in E^S} f_{ij}^S - \sum_{j:e_{ji}^R \in E^R} f_{ji}^R \right) = 0$$

where the last equality follows from that f' and f are circulation flows in G. Therefore, g is a circulation flow in $\overline{G}$. We observe that the flow on any edge $e \in \overline{E}^R \setminus E_f^R$ is zero in g because 1) if $e = \overrightarrow{e}_{ij}^R$, then we have $f_{ij}^R = R_{ij}$ and hence $\overrightarrow{g}_{ij}^R = 0$; 2) if $e = \overleftarrow{e}_{ij}^R$, then we have $f_{ij}^R = 0$ and hence $\overleftarrow{g}_{ij}^R = 0$. We further observe that $\overrightarrow{g}_{ij}^S \leq 0$ for any edge $\overrightarrow{e}_{ij}^S \in \overline{E}^S \setminus E_f^S$ since we have $f_{ij}^S = S_{ij}$. Since $\overrightarrow{W}_{ij}^S = 0, \forall \overrightarrow{e}_{ij}^S \in \overline{E}^S$, the weight of flow g in $\overline{G}$ is

$$\sum_{i,j:\overrightarrow{e}_{ij}^R \in \overline{E}^R} \left(\overrightarrow{W}_{ij}^R \overrightarrow{g}_{ij}^R + \overleftarrow{W}_{ij}^R \overleftarrow{g}_{ij}^R \right) = \sum_{i,j:e_{ij}^R \in E^R} U_{ij} \left(f_{ij}'^R - f_{ij}^R \right)$$
$$= \sum_{i,j:e_{ij}^R \in E^R} U_{ij} f_{ij}'^R - \sum_{i,j:e_{ij}^R \in E^R} U_{ij} f_{ij}^R > 0$$

where the last inequality follows from the assumption that f' has larger utility than f in G. Since g only has positive flows on the edges in G_f, using a similar argument as in the proof of Lemma 1, g is the aggregate flow of balanced flows along cycles each comprised of edges in G_f. Then the total weight of these flows along the cycles in G_f is equal to the weight of flow g in $\overline{G}$, which is greater than 0. This implies that there must exist a cycle of positive weight in G_f, which is a contradiction to the previous assumption. This completes the proof.

Proof of Theorem 3

As discussed earlier, we first equivalently convert the problem to one with integral parameters by multiplying them by an integer K.

Since the capacities of all edges in the graph are integral and the initial empty flow is integral, the residual capacity of the cycle found in the first iteration of the algorithm is integral, and hence the flow after augmentation is integral. Thus, by induction, the updated flow after each iteration is also integral. This shows that the algorithm finds an integral flow when it terminates, which is optimal by Lemma 2.

The utility of the initial empty flow is 0. The utility of any flow is upper bounded by the utility of the flow that saturates all request edges, which is $|E^R|\overline{RU}K^2$. Since the capacities of all edges are integral, the flow utility increases by an integer no less than one at each iteration of Algorithm 1. Therefore, it takes the algorithm at most $|E^R|\overline{RU}K^2$ iterations to terminate. Since each iteration has running time $O(|V|(|E^S| + |E^R|)$, the desired result follows.