A Market Clearing Solution for Social Lending

Ning Chen[†] **Arpita Ghosh**[‡]
[†]Division of Mathematical Sciences, Nanyang Technological University, Singapore.
ningc@ntu.edu.sg
[‡]Yahoo! Research, Santa Clara, CA, USA.
arpita@yahoo-inc.com

Abstract

The social lending market, with over a billion dollars in loans, is a two-sided matching market where borrowers specify demands and lenders specify total budgets and their desired interest rates from each acceptable borrower. Because different borrowers correspond to different risk-return profiles, lenders have preferences over acceptable borrowers; a borrower prefers lenders in order of the interest rates they offer to her. We investigate the question of what is a computationally feasible, 'good', allocation to clear this market.

We design a strongly polynomial time algorithm for computing a Pareto-efficient stable outcome in a two-sided many-to-many matching market with indifferences, and use this to compute an allocation for the social lending market that satisfies the properties of stability — a standard notion of fairness in two-sided matching markets — and Pareto efficiency; and additionally addresses envy-freeness amongst similar borrowers and risk diversification for lenders.

1 Introduction

Social lending, or peer-to-peer lending, which allows individuals to lend and borrow money to each other directly without the participation of banks, is an exploding business on the Internet: the total amount of money borrowed using such peer-to-peer loans was approximately $\$650$ million in 2007, and is projected to reach $\$5.8$ *billion* by 2010.

The social lending market consists of borrowers seeking some target loan amount (their demand), and lenders seeking to invest some fixed amount of money in loans (their budget). Lenders usually prefer to invest their budget in multiple borrowers' loans to spread risk from defaulting borrowers, and each borrower's loan is also usually funded by multiple lenders. Borrowers are 'non-homogeneous' — different borrowers have different characteristics such as credit rating and desired loan length, and command different interest rates based on their creditworthiness. That is, different borrowers correspond to investments with different risk-return profiles. As a result, lenders have *preferences* over which borrowers they would like to lend their money to — one lender may prefer high-risk-high-return borrowers, while another may prefer safe borrowers albeit fetching lower interest rates. On the other hand, borrowers also have implicit preferences over lenders since different lenders can offer different interest rates to the same borrower: a borrower simply prefers the lenders who offer him the lowest rate.

We now have a large two-sided matching market where agents on both sides have multiunit capacities, and preferences — lenders have preference rankings (possibly with ties) over the set of acceptable borrowers they're willing to lend to, and borrowers have preferences over lenders based on their offered interest rates. But the preferences of all these agents may be conflicting — many lenders may compete to lend to the same borrower who is their common top choice, while this borrower's preference might be an entirely different lender who offers him a lower interest rate, who in turn has a different top borrower choice. Clearly, it need not be possible to make all agents simultaneously happy, raising the natural question of what constitutes a 'good' assignment. In this paper, we investigate the social lending market from a computational social choice perspective: what is a fair and efficient way to clear this marketplace, and how can it be computed?

Our contributions. We present a model (§2) for the social lending marketplace based on Zopa (www.zopa.com), which, with over $400,000$ members and $\pounds 100$ million in its markets and over $\pounds 100000$ traded *each day*, is the first and among the largest social lending sites on the Web.

We first address the question of what is a desirable allocation in the social lending market in §3, and argue that an allocation that is stable, Pareto efficient, fair amongst 'equal' borrowers, and also addresses the need for risk diversification to reduce default risk is a desirable solution concept in this market. We then address the question of finding an algorithm that returns such an allocation in §4.

When preference lists contain ties, as in our social lending context, not all stable matchings are Pareto efficient. The question of how to find a Pareto-stable matching when preferences contain ties has recently been addressed for the *many-to-one matching* model in [Erdil and Ergin, 2008; 2009]. A naive adaptation of this algorithm to our *many-to-many* market returns a Pareto-stable assignment in time that scales with the total capacity of all nodes in the graph, *i.e.*, the amount of money traded in the market, requiring us to de-

velop a new algorithmic approach. We design a strongly polynomial time algorithm for computing a Pareto-stable outcome in a many-to-many matching market with indifferences, and apply it to a reduced marketplace where 'identical' borrowers are grouped into equivalent classes or 'categories'; we then reallocate amongst categories to achieve fairness amongst borrowers and risk diversification. The overall runtime is polynomial in the number of lenders and the number of borrower categories, which is a small constant (=10) for the Zopa marketplace.

Related work. Social lending is a relatively new application that has only recently begun to be addressed in the research literature, starting with the work in [Freedman and Jin, 2009] on default rates. Most of the research on social lending takes an empirical approach as in [Freedman and Jin, 2009]; [Chen *et al.*, 2009] analyzes the auction held for a single borrower's loan in the Prosper market, but does not address the several coexisting lenders and borrowers in the marketplace. To the best of our knowledge, social lending has not been studied much from a marketplace design or social choice perspective.

There is a very vast literature on two-sided matching markets and stable matching; for a review of the economics literature on the subject, see [Roth and Sotomayor, 1992; Roth, 2008]; for an introduction to the computer science literature addressing algorithmic and computational questions, see, e.g., [Gusfield and Irving, 1989; Iwama and Miyazaki, 2008]. The paper most relevant to our work from the stable matching literature is [Erdil and Ergin, 2008; 2009], who study the algorithmic question of finding Pareto-stable matchings for a *many-to-one* matching market; see §4. The many-to-many setting is far less well studied in the stable matching literature, and focuses largely on structural results in settings without indifferences; see, e.g., [Hatfield and Kominers, 2010; Echenique and Oviedo, 2006].

2 A Lending Market Model

We model the social lending marketplace M as a bipartite graph with nodes (A, B) and edges E. The nodes in A represent the lenders and the nodes in B are the borrowers. Nodes on both sides have multiunit capacities: a lender i's capacity c_i is her budget, the total amount of money she wants to lend. A borrower j's capacity c_j is his demand, the total loan amount he wants to borrow. We will assume that the capacities are integers by expressing them in the smallest unit of currency. The edge set E of M is the set of pairs (i, j) where lender i is willing to lend to borrower j.

Each lender specifies the interest rates at which she is willing to lend money to different acceptable borrowers; as in Zopa, this is the actual interest-rate that she will receive on any loans to that borrower. Note that each lender can offer different interest rates to different borrowers, and the same borrower can be offered different interest rates by different lenders.

Every acceptable borrower, along with the specified interest rate, represents a possible investment for the lender, with a particular risk-return profile. Each lender has a *preference* list P_i ranking the investments corresponding to these borrower-interest rate pairs, *i.e.*, its neighbors $\{j \in B : (i, j) \in E\}$. These preferences need not be strict and the lender can be indifferent between, *i.e.*, equally prefer, two different investments; that is, preference lists can have ties. Since the preference list is restricted to i's neighbors (*i.e.*, acceptable borrowers), it is naturally incomplete. (As an example, lender i's preferences, denoted $P_i = ([j_1, j_2], [j_3, j_4, j_5])$, could be as follows: i is indifferent between j_1 and j_2, and prefers both of them to j_3, j_4, j_5 among all of whom i is indifferent; she finds all other borrowers unacceptable.) In general, a lender can also have preferences over sets of investments; however, here we will restrict ourselves to expressing preferences over individual investments for simplicity[1].

Each borrower j has an implicit *preference* ranking P_j over lenders based on the interest rates they offer him: j prefers lenders in non-increasing order of offered interest rates, and is indifferent amongst lenders who offer him the same interest-rate (so borrowers' preferences P_j can contain ties as well).

We partition the set of borrowers into equivalence classes, or *categories* $C = \{C_1, \ldots, C_m\}$: two borrowers are equivalent, *i.e.*, belong to the same category, if no lender can distinguish between them based on the information available about them in the marketplace. Thus, all lenders are indifferent between the borrowers in a category, and offer them the same interest-rate. This also means that borrowers in a category all have the same preferences over lenders, and that a lender's preference list need only rank categories, not individual borrowers. The number of categories can be as large as the number of borrowers when *personal* information is provided by/about each borrower (as in Prosper), or very small when the only information revealed is the credit-rating and loan length (as in Zopa, which only allows lenders to specify interest rates for 10 borrower categories).

Diversification to decrease default risk is a very important factor in social lending. Instead of modeling this into the preferences of lenders, we deal with it as Zopa— Zopa breaks up each lender's budget into small sums each of which is lent to a different borrower to diversify risk, so we will similarly ensure that each lender's budget is uniformly spread amongst many different borrowers in the final allocation.

We note that we do not model reserve rates, nor the dynamic aspect of the social lending market in this work.

Feasible assignments. The output of the market is a multiunit pairing, or *assignment* $X = (x_{ij})_{(i,j)\in E}$ between A and B, where $x_{ij} \in \mathbb{N} \cup \{0\}$ is the number of units assigned from $i \in A$ to $j \in B$ (when $c_i = c_j = 1$ for all nodes, an assignment reduces to a *matching*). An assignment X is *feasible* if it simply satisfies capacity constraints on both sides, *i.e.*, $\sum_j x_{ij} \leq c_i$ and $\sum_i x_{ij} \leq c_j$. Note that the preferences P_i, P_j do not matter to the feasibility of an assignment.

3 What Is a Good Outcome?

Having defined the set of feasible assignments, how do we choose one from amongst the very large number of possible assignments? An ideal solution concept for the social lending market would be Pareto efficient, fair — both across lenders

[1]This is both for technical tractability and to avoid eliciting complex combinatorial preferences from lenders.

and borrowers, as well as amongst similar borrowers — *exist* for every instance of the input, and be efficiently computable (since social lending markets transact huge amounts of money, it should be implementable in time that depends only on the number of agents, not on the money being traded in the marketplace). What assignment has these properties?

A very widely used solution concept in two-sided matching markets is that of *stability* [Gale and Shapley, 1962]: there is no pair of individuals that *both* strictly prefer each other to some partner they are currently assigned to (such a pair would be called a blocking pair). Stability can be interpreted as a notion of fairness in our context — while it is not possible to guarantee each lender her most preferred allocation, a stable allocation is fair in the sense that if a lender indeed sees a better allocation, that allocation does not also 'prefer' her in return. However, when preference lists contain ties as in our case, it is well-known that stable matchings need not be *Pareto efficient*, even when all nodes have unit capacity; see [Roth and Sotomayor, 1992].

Therefore, we will need to explicitly require that the solution is both stable and Pareto efficient; such assignments are called *Pareto-stable* assignments [Sotomayor, 2009]. However, as the next example shows, applying the concept of Pareto stability directly to the marketplace M may not produce very desirable solutions: a solution may well be Pareto-stable, but hand out very different interest rates to two identical borrowers, violating our fairness requirement.

Example 3.1. *There are two lenders i_1, i_2 and two borrowers j_1, j_2 with two units of supply/demand each. Both lenders are indifferent between the two borrowers; the first lender i_1 offers 7% to both borrowers, and the second lender i_2 offers 15% to both borrowers. The matching where i_1 lends both units to j_1 (at 7%), and i_2 lends both units to j_2 (at 15%) is stable and also Pareto efficient. However, the matching where both lenders lend one unit to each borrower is also Pareto-stable and 'more fair' since both j_1 and j_2 get equal amounts of the low and high interest rates (note that this matching does not Pareto-improve the previous matching since it makes j_1 strictly worse off). In addition, each lender spreads her loan across more borrowers, so diversity improves as well.*

This example illustrates that we cannot simply apply the solution concept of Pareto stability directly to the social lending marketplace M. Instead, we will consider a modified market, where the lender side is unchanged but borrowers are aggregated by category into 'meta-borrowers', with demand equal to the aggregate demand of that category (recall that a category can consist of a single borrower, when there is plenty of information about borrowers in the marketplace). We will first find a Pareto-stable assignment in this *reduced* marketplace— how to find such an assignment is the key technical problem we need to solve— and then distribute each lender's allocation to a meta-borrower amongst all the borrowers in that category to ensure envy-freeness amongst them.

We note that another natural solution concept, maximum size assignment (*i.e.*, the one with the largest trade volume), is unsuitable here since it ignores all agents' preferences; it is also NP-hard to compute [Iwama *et al.*, 1999]. However, it is easy to show that the size of any stable assignment (and therefore also our assignment) is at least half the size of the maximum size assignment.

The assignment we propose to clear the social lending market is the following.

MARKET CLEARING ASSIGNMENT: Given a lending market $M = (A, B)$ with categories $C = \{C_1, \ldots, C_m\}$:

1. Create a meta-borrower for each category C_r with demand $\sum_{j \in C_r} c_j$, and the same preferences as those of borrowers in C_r. Denote the resulting market by (A, C).
2. Compute a Pareto-stable assignment $X^* = (x^*_{iC_r})$ for (A, C), where $x^*_{iC_r}$ is lender i's total investment in category C_r (§4).
3. Assign each lender's investment $x^*_{iC_r}$ across all borrowers in category C_r to ensure diversity and envy-freeness; denote the final assignment by $Y = (y_{ij})$ (§4).

We have the following result about this assignment.

Theorem 3.2 (Main). *The final assignment $Y = (y_{ij})$ can be computed in time $O(|A|^4 + |A||B|)$ and has the following properties:*

1. *Stability: There are no blocking pairs in the original marketplace $M = (A, B)$.*
2. *Pareto efficiency: No agent in M can be made better off without making some other agent in M strictly worse off.*
3. *(Weak) envy-freeness: No borrower envies the allocation of any other borrower in its category.*
4. *Diversity: Given the allocations $X^* = (x^*_{iC_r})$, each lender i spreads her budget amongst the maximum number of distinct borrowers.*

4 Algorithm

We will first address the problem of efficiently finding a Pareto stable assignment in an abstract two-sided many-to-many matching market with separable responsive preferences, and then apply the algorithm we develop to the modified marketplace with lenders and meta-borrowers.

We begin with some formal definitions. Recall that we have a two-sided matching market $M = (A, B)$ with preference lists P_k and multi-unit capacities c_k for all agents k, and k's preference over sets is the natural (partial) order defined by the preferences P_k over individuals as in [Erdil and Ergin, 2008; 2009]. We can assume without loss of generality that $|A| = |B| = n$ by adding dummy isolated nodes with $c_k = 0$ to the market.

Definition 4.1 (Level function). *We use the function $L_i(\cdot)$ to encode the preference list of a node $i \in A$. For each $j \in P_i$, let $L_i(j) \in \{1, \ldots, n\}$ denote the* ranking *of j in i's preference list. Therefore, for any $j, j' \in P_i$, if $L_i(j) < L_i(j')$, then i strictly prefers j to j'; if $L_i(j) \leq L_i(j')$, then i weakly prefers j to j'; if $L_i(j) = L_i(j')$, then i is indifferent between j and j'. The definition of the level function $L_j(\cdot)$ for each $j \in B$ is symmetric.*

Stability. We say that an assignment $X = (x_{ij})$ is stable if there is no blocking pair (i, j), $i \in A$ and $j \in B$, $(i, j) \in E$, such that both i and j have leftover capacity; *or* i has leftover capacity and there is i', $x_{i'j} > 0$, such that j strictly prefers i to i' (or similarly for some j); *or* there are i' and j', $x_{ij'} > 0$ and $x_{i'j} > 0$, such that i strictly prefers j to j' and j strictly prefers i to i'. Note that *both* members of a blocking pair must strictly prefer to trade with each other. A stable assignment always exists, and can be found (efficiently) using a variant of Gale-Shapley algorithm [Gale and Shapley, 1962] for computing stable matchings.

Pareto efficiency. Given an assignment $X = (x_{ij})$, let $x_i(\alpha) = \sum_{j:\ L_i(j) \le \alpha} x_{ij}$ be the number of units of i's capacity that is assigned at levels no worse than α, and $x_j(\beta) = \sum_{i:\ L_j(i) \le \beta} x_{ij}$ be the number of units of j's capacity that is assigned at levels no worse than β. We say that X is Pareto efficient if there is no other feasible assignment $Y = (y_{ij})$ such that $y_i(\alpha) \ge x_i(\alpha)$ and $y_j(\beta) \ge x_j(\beta)$, for all i, j and α, β, and at least one of the inequalities is strict. That is, X is not Pareto-dominated by any other assignment where at least one agent is strictly better off and no one is worse off.

Pareto stability. A feasible assignment is called *Pareto-stable* if it is both stable and Pareto efficient.

Recall that when preference lists contain ties, a stable matching need not be Pareto efficient. The following definition is critical to our algorithm for Pareto-stable assignment.

Definition 4.2 (Augmenting Path and Cycle). *Given an assignment $X = (x_{ij})$, a sequence $[i_0, j_1, i_1, \ldots, j_\ell, i_\ell, j_{\ell+1}]$ is an* augmenting path *if the following conditions hold:*

- $x_{i_0} < c_{i_0}$ *and* $x_{j_{\ell+1}} < c_{j_{\ell+1}}$.
- $x_{i_k j_k} > 0$ *for* $k = 1, \ldots, \ell$.
- $L_{i_k}(j_k) \ge L_{i_k}(j_{k+1})$ *and* $L_{j_k}(i_{k-1}) \le L_{j_k}(i_k)$ *for* $k = 1, \ldots, \ell$.

A sequence $[i_1, j_2, i_2, \ldots, j_\ell, i_\ell, j_1, i_1]$ is an augmenting cycle *if the following conditions hold:*

- $x_{i_k j_k} > 0$ *for* $k = 1, \ldots, \ell$.
- $L_{i_k}(j_k) \ge L_{i_k}(j_{k+1})$ *and* $L_{j_k}(i_{k-1}) \le L_{j_k}(i_k)$ *for* $k = 1, \ldots, \ell$, *where* $i_0 = i_\ell$ *and* $j_{\ell+1} = j_1$.
- *At least one of these inequalities is strict. If i_k is such a node, we say the augmenting cycle is associated with i_k at level $L_{i_k}(j_k)$ (and similarly for j_k.)*

Since our nodes have preferences in addition to capacities, augmenting paths and cycles must *improve not just the size of an assignment but also its quality*, as given by node preferences. The first condition in the definition of the augmenting path says that the capacities of i_0 and $j_{\ell+1}$ are not exhausted. The second condition says that there is a positive allocation from i_k to j_k in the current assignment X, and the last condition says that i_k weakly prefers j_{k+1} to j_k and j_k weakly prefers i_{k-1} to i_k. Thus, we can inject (at least) one unit of flow from i_{k-1} to j_k and from i_ℓ to $j_{\ell+1}$ and withdraw the same amount from i_k to j_k for each $k = 1, \ldots, \ell$ in the augmenting path to obtain a Pareto improvement over X. A similar Pareto improvement can be obtained for augmenting cycles.

We have the following easy lemma.

Lemma 4.3. *Any feasible assignment X that has no augmenting paths or cycles is Pareto efficient.*

4.1 Computing a Pareto Stable Assignment

We now give a strongly polynomial time algorithm to compute a Pareto stable assignment. Note that if X is a stable assignment, reassigning according to any augmenting path or cycle of X preserves stability, *i.e.*, any assignment Y that Pareto-dominates a stable assignment X is stable as well [Erdil and Ergin, 2009]. Together with Lemma 4.3, this suggests that starting with a stable assignment, and then making improvements to it using augmenting paths and cycles until no more improvements are possible, will result in a Pareto-stable assignment.

How do we find such augmenting paths and cycles? First consider the simplest case with unit capacity, *i.e.*, $c_i = c_j = 1$ for all i, j, where an assignment degenerates to a matching. Given an existing matching, define a new directed bipartite graph with the same nodes, where all forward edges are "weak improvement" edges with respect to the existing matching, and backward edges correspond to the pairings in current matching. Then we can find augmenting paths by introducing a source s and sink t that link to unmatched nodes on each side and finding s-t paths in the resulting network. Augmenting cycles can be found by a similar construction.

For our general case where $c_i, c_j \ge 1$, however, *even the concept of improvement edges for a node is not well defined*: since a node can have multiple partners in an assignment, a particular edge can be an improvement for some part of that node's capacity and not for some others. For instance, suppose that node i (with $c_i = 2$) is matched to nodes j_1 and j_3, and suppose that i strictly prefers j_1 to j_2 to j_3. Then, (i, j_2) would only represent an improvement relative to (i, j_3), but not with respect to (i, j_1), both of which exist in the current assignment. An obvious way to fix this problem is to make copies of each node, one copy for each unit of its capacity, in which case improvement edges are well-defined — each unit of capacity is associated with a unique neighbor in any assignment. However, this new graph has size $\sum_i c_i + \sum_j c_j$, leading to a runtime that is polynomial in $\sum_i c_i + \sum_j c_j$, which is *exponential* in the size of the input.

Construction of networks. In order to define improvement edges in this setting with multiunit capacities, we will create a new augmented bipartite graph G from the original bipartite market M and the preference lists P_k. The vertex set of G will consist of copies of each node in M, where *each copy represents a level on that node's preference list*. We then define forward and backward edges between the vertices: forward edges are the (weak) improvement edges, while there is one backward edge for every edge $(i, j) \in E$ corresponding to i and j's levels in P_j and P_i. This augmented graph, which is assignment-independent and depends only on the preference lists of the nodes, is then used to define a sequence of networks with assignment-dependent capacities which we will use to find augmenting paths and cycles.

Definition 4.4. *Given the market M, construct G as follows.*

- *Vertices: For each node $i \in A \cup B$, we introduce n new vertices $T(i) = \{i(1), \ldots, i(n)\}$, where $i(\alpha)$ cor-*

responds to the α-th level of the preference list of i. (If i has $k < n$ levels in his preference list, it suffices to introduce k vertices $i(1), \ldots, i(k)$; here, we use n levels for uniformity.)

- *Edges: For each pair $(i, j) \in E$, let $\alpha = L_i(j)$ and $\beta = L_j(i)$. We add a* backward *edge between $i(\alpha)$ and $j(\beta)$, i.e., $j(\beta) \to i(\alpha)$. Further, we add a* forward *edge $i(\alpha') \to j(\beta')$ for every pair of vertices $i(\alpha')$ and $j(\beta')$ satisfying $\alpha' \geq \alpha$ and $\beta' \geq \beta$.*

Figure 1 gives an example of the construction of graph G, when M contains two lenders i_1, i_2 and three borrowers j_1, j_2, j_3 (node preferences are specified next to each node in the top figure, e.g., i_2 is indifferent between j_1 and j_2, and prefers both of them to j_3). The figure on the right illustrates the vertices $T(\cdot)$ of G and the backward edges; the figure on the bottom left shows the forward edges between the two groups of vertices $T(i_2)$ and $T(j_3)$ in G.

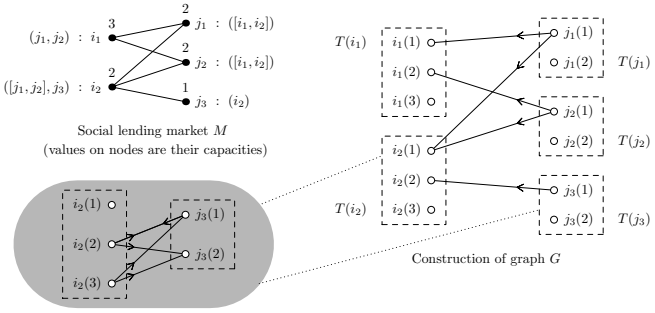

Figure 1: Construction of graph G.

Note that the construction of G is completely independent of any actual assignment X. We next define the networks $H, H_{i,\alpha}, H_{j,\beta}$, whose structure is based on G and is assignment-independent, but whose edge capacities depend on the assignment X.

Definition 4.5 (Network H, $H_{i,\alpha}$ and $H_{j,\beta}$). *Given the graph G and an assignment $X = (x_{ij})$, let $G(X)$ be the network where all forward edges in G are assigned capacity ∞, and all backward edges are assigned capacity x_{ij}. We use $G(X)$ to define the networks $H(X)$, $H_{i,\alpha}(X)$ and $H_{j,\beta}(X)$ for each $i \in A$ and $j \in B$, and $\alpha, \beta = 1, \ldots, n$, as follows.*

For $H(X)$, include a source s and a sink t; further, for each $i \in A$ and $j \in B$, add an extra vertex h_i and h_j, respectively. Connect $s \to h_i$ with capacity $c_i - x_i$, and $h_j \to t$ with capacity $c_j - x_j$, where $x_i = \sum_j x_{ij}$ and $x_j = \sum_i x_{ij}$. Further, connect $h_i \to i(\alpha)$ with capacity ∞ for $\alpha = 1, \ldots, n$, and connect $j(\beta) \to h_j$ with ∞ capacity for $\beta = 1, \ldots, n$.

For $H_{i,\alpha}(X)$, we add a source s and a sink t, and connect $s \to j(\beta)$ with capacity ∞ for each vertex $j(\beta)$ satisfying $\alpha > L_i(j)$ and $\beta \geq L_j(i)$. Further, we connect $j(\beta) \to t$ with capacity x_{ij} for each $j(\beta)$ satisfying $\alpha \leq L_i(j)$ and $\beta = L_j(i)$. The network $H_{j,\beta}(X)$ is defined symmetrically.

We will use the network H to find augmenting paths with respect to an existing stable assignment X. Observe that the only edges from the source with nonzero capacity are those that connect to a node $i \in A$ with leftover capacity; similarly, the only edges to the sink with nonzero capacity are from a node $j \in B$ with leftover capacity. Sending flow from s to t in H therefore involves increasing the total size of the assignment while maintaining quality, exactly as in an augmenting path for X. Similarly, we will use the networks $H_{i,\alpha}$ and $H_{j,\beta}$ to find augmenting cycles associated with i and j at level α and β respectively. Consider any flow from s to t in $H_{i,\alpha}$, say,

$$[s, j_1(\beta_1), i_1(\alpha_1), \ldots, i_2(\alpha_2), j_2(\beta_2), t]$$

We know that $\alpha > L_i(j_1)$ (*i.e.*, i strictly prefers j_1 to all its neighbors at level α) and $L_{j_1}(i_1) = \beta_1 \geq L_{j_1}(i)$ (*i.e.*, j_1 weakly prefers i to i_1). Further, we have $\alpha \leq L_i(j_2)$ (this implies that i strictly prefers j_1 to j_2) and $L_{j_2}(i_2) \leq \beta_2 = L_{j_2}(i)$ (*i.e.*, j_2 weakly prefers i_2 to i). That is, flows from s to t in $H_{i,\alpha}$ correspond to augmenting cycles for node i at levels less than or equal to α in X (and similarly for $H_{j,\beta}$).

Our algorithm, summarized below, finds maximum flows in all the constructed networks H, $H_{i,\alpha}$ and $H_{j,\beta}$ to eliminate augmenting paths and cycles.

PARETO STABLE ASSIGNMENT (ALG-PS)

1. Let X be an arbitrary stable assignment
2. Construct networks $H(X)$, $H_{i,\alpha}(X)$ and $H_{j,\beta}(X)$, for each $i \in A$, $j \in B$, and $\alpha, \beta = 1, \ldots, n$
3. For H, $H_{i,\alpha}$ and $H_{j,\beta}$ constructed above (H to be executed first)
   (a) Compute a maximum flow $F = (f_{uv})$ from s to t (if there is no flow from vertex u to v, set $f_{uv} = 0$)
   (b) For each forward edge $i(\alpha) \to j(\beta)$, let $x_{ij} = x_{ij} + f_{i(\alpha)j(\beta)}$
   (c) For each backward edge $j(\beta) \to i(\alpha)$, let $x_{ij} = x_{ij} - f_{j(\beta)i(\alpha)}$
   (d) If the graph is $H_{i,\alpha}$
   - Let $x_{ij} = x_{ij} + f_{sj(\beta)}$ for each $s \to j(\beta)$
   - Let $x_{ij} = x_{ij} - f_{j(\beta)t}$ for each $j(\beta) \to t$

   (e) If the graph is $H_{j,\beta}$
   - Let $x_{ij} = x_{ij} - f_{sj(\beta)}$ for each $s \to i(\alpha)$
   - Let $x_{ij} = x_{ij} + f_{j(\beta)t}$ for each $i(\alpha) \to t$

   (f) Update capacities for next graph to be executed according to new assignment X
4. Output X (denoted by X^*)

Analysis. To prove that ALG-PS indeed computes a Pareto-stable assignment, we need to show two things— first, that the assignment X^* returned by the algorithm is stable; this follows easily from stability of the original assignment and that reassigning according to augmenting paths and cycles preserves stability.

Second, we need to show that X^* is Pareto efficient, *i.e.*, no further Pareto improvements are possible when the algorithm terminates. The difficulty here is that the assignment X changes through the course of the algorithm, and therefore we need to show that, for instance, no other augmenting paths can be found after the network H has been executed, even

though the assignment X that was used to define the network $H(X)$ has been changed (and similarly for all augmenting cycles). That is, while we compute maximum flows in $H(X)$ to find all augmenting paths for a given assignment X, we need to show that no new augmenting paths have showed up in any updated assignments computed by the algorithm. Similarly, finding (i, α) augmenting cycles via $H_{i,\alpha}(X)$ for *some* assignment X does not automatically imply that no further (i, α) augmenting cycles will ever be found in *any* of the (different) assignments computed through the course of the algorithm, since the assignments of all nodes can change each time when a maximum flow is computed, leading to the possibility of new valid s-t paths, and therefore possibly new augmenting cycles. That this does not is due to a careful choice of the construction of the networks H, $H_{i,\alpha}$, $H_{j,\beta}$; in fact, it is possible to construct examples showing that this does not hold for other, perhaps more natural, definitions of the networks.

Our main claim is stated next. The proof uses the assignment-independence of the structure of the networks $H, H_{i,\alpha}, H_{j,\beta}$ to argue that if there is an augmenting path in any assignment produced after H is executed, we could not have found the maximum flow in $H(X)$ to begin with, a contradiction; the argument for augmenting cycles uses a similar idea. All proofs can be found in the full version of the paper[2].

Proposition 4.6. *There is no augmenting path after graph H is executed, and no augmenting cycle associated with i (resp. j) at level α (resp. β) after graph $H_{i,\alpha}$ (resp. $H_{j,\beta}$) is executed in* ALG-PS.

The above claim, together with Lemma 4.3, implies that the outcome returned by ALG-PS is indeed a Pareto-efficient assignment as required.

Running time. Each graph $H, H_{i,\alpha}$ and $H_{j,\beta}$ can be constructed in time $O(m^2n^2)$, where $n = |A|$ and $m = |B|$, and there are $O(mn)$ such graphs in all. Each graph has $O(mn)$ vertices, and is executed exactly once in time $O(m^3n^3)$, which is the running time for maximum flow using any classic network flow algorithm. Therefore, the running time of the algorithm is in $O(m^4n^4)$. We summarize this below.

Theorem 4.7. *Algorithm* ALG-PS *computes a Pareto-stable assignment in strongly polynomial time $O(m^4n^4)$.*

We note that the number of borrower categories in Zopa is a small constant, so this algorithm computes a Pareto-stable assignment in our reduced marketplace (A, C) in time $O(n^4)$ where $n = |A|$ is the number of lenders. In fact, a sharper bound on the running time of the algorithm is $O((\sum_{k\in M} |P_k|)^4)$, where $|P_k|$ is the length of k's preference list. This means that even when each category contains a single borrower as in Prosper (so m is large), the runtime remains practically feasible: since lenders usually place bids on only a small number of borrowers in typical social lending markets, $\sum_{k\in M} |P_k| = O(n)$ leading to runtime $O(n^4)$.

Computing the Market Clearing Assignment. Having computed allocations $X^* = (x^*_{iC_r})$ between lenders and borrower categories using algorithm ALG-PS, we now need to allocate the amount $x^*_{iC_r}$ amongst borrowers in C_r. Note that by feasibility of X^* for (A, C), we have $\sum_{i\in A} x^*_{iC_r} \leq \sum_{j\in C_r} c_j$. We simply divide $x^*_{iC_r}$ amongst borrowers in C_r proportional to their demands:

$$y_{i_0 j_0} = x^*_{i_0 C_r} \cdot \frac{c_{j_0}}{\sum_{j\in C_r} c_j}. \qquad (*)$$

This allocation is feasible since $\sum_{j\in C_r} y_{ij} = x^*_{iC_r}$ and $\sum_{i\in A} y_{ij} \leq c_j$. This assignment $Y = (y_{ij})$ can be proven to satisfy all the properties claimed in Theorem 3.2 for the actual marketplace $M = (A, B)$, and is our desired output.

Acknowledgements. We are very grateful to Gabrielle Demange, Bettina Klaus, Fuhito Kojima, Mohammad Mahdian, Preston McAfee, David Pennock, Michael Schwarz and anonymous referees for helpful discussions and comments.

References

[Chen *et al.*, 2009] N. Chen, A. Ghosh, and N. Lambert. Social lending. In *EC 2009*, pages 335–344, 2009.

[Echenique and Oviedo, 2006] F. Echenique and J. Oviedo. A theory of stability in many-to-many matching markets. *Theoretical Economics*, 1:233–273, 2006.

[Erdil and Ergin, 2008] A. Erdil and H. Ergin. What's the matter with tie-breaking? Improving efficiency in school choice. *American Economic Review*, 98:669–689, 2008.

[Erdil and Ergin, 2009] A. Erdil and H. Ergin. Two-sided matching with indifferences. Working paper. 2009.

[Freedman and Jin, 2009] S. Freedman and G. Jin. Learning by doing with asymmetric information: Evidence from Prosper.com. Working paper. 2009.

[Gale and Shapley, 1962] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, pages 9–15, 1962.

[Gusfield and Irving, 1989] D. Gusfield and R. W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, 1989.

[Hatfield and Kominers, 2010] J. Hatfield and S. Kominers. Matching in networks with bilateral contracts. In *EC 2010*, pages 119–120, 2010.

[Iwama and Miyazaki, 2008] K. Iwama and S. Miyazaki. *Stable Marriage with Ties and Incomplete Lists*. Encyclopedia of Algorithms, 2008.

[Iwama *et al.*, 1999] K. Iwama, D. Manlove, S. Miyazaki, and Y. Morita. Stable marriage with incomplete lists and ties. In *ICALP 1999*, pages 443–452, 1999.

[Roth and Sotomayor, 1992] A. E. Roth and M. Sotomayor. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Cambridge University Press, 1992.

[Roth, 2008] A. E. Roth. Deferred acceptance algorithms: History, theory, practice, and open questions. *International Journal of Game Theory*, pages 537–569, 2008.

[Sotomayor, 2009] M. Sotomayor. Pareto-stability is a natural solution concept in matching markets with indifferences. Working paper. 2009.

[2] http://www.ntu.edu.sg/home/ningc/paper/ijcai11-z.pdf