**Βήμα 1:** Ανάθεση βάρους στην επιλογή του συνεργάτη jγια τον i:

Στις περιπτώσεις 1 και 2 (όπου οι δύο έχουν ξανασυνεργαστεί) η βαθμολογία κλιμακωτά για να συνυπολογίσει τις εξής περιπτώσεις:

1. χρηστών που έχουν βολευτεί σε μία μέτρια κατάσταση (μπας και τους ξεκουνήσει προς κάτι καλύτερο – ή χειρότερο), π.χ. βαθμολογώ κάποιον με 3, αλλά λέω ότι θέλω να ξανασυνεργαστώ. Τότε το βάρος προτίμησης θα γίνει 1.6 αντί για 2.
2. Χρηστών που ναι μεν τους έχω βαθμολογήσει θετικά, αλλά λέω «ας δούμε για κανέναν άλλο». Έτσι αν έχω βαθμολογήσει κάποιον με 4 και τώρα λέω «δεν θέλω να συνεργαστώ ξανά», η βαρύτητα θα είναι -1.2 αντί για -2.

Φυσικά οι κλίμακες μπορούν να αλλάξουν, όπως και τα μοντέλα που λαμβάνονται υπ’ όψιν παλιότερες βαθμολογίες (π.χ. οι πιο πρόσφατες θα μπορούσαν να λαμβάνονται ισχυρότερα υπ΄ όψιν σε σχέση με τις πιο παλιές) ή η κατανομή βάρους ανάμεσα στο κριτήριο «μη συνεργάσιμος» και στο κριτήριο «δεν γράφει καλά».

**Βήμα 2:**Utility functionγια κάθε χρήστη

όπου xi,j μια δυαδική μεταβλητή που παίρνει τιμές 0 ή 1: 1🡪 ανατίθεται ο j ως partner του ι, 0 🡪δεν ανατίθεται ο j ως partner του ι

Παράλληλα εισάγεται ένας περιορισμός

(προφανώς: μόνο ένας partnerμπορεί να ανατεθεί σε κάθε χρήστη)

**Βήμα 3:** Συνολικό utility function:

Και πρόσθετοι περιορισμοί:

xi,j=xj,i, ∀i, j

Πιθανώς να χρειάζεται και:

(για κάθε player, θα υπάρχει ένας που θα τον έχουμε ταιριάξει, αν και μάλλον προκύπτει από τα παραπάνω)

**Βήμα 4:**

Λύνουμε το πρόβλημα, μεγιστοποιώντας ως προς το . Η λύση είναι το σύνολο αναθέσεων τιμών στις μεταβλητές xi,j.

**Πιθανή επέκταση:**

Για τις περιπτώσεις όπου wi,j<THRESHOLD μπορούμε να βάλουμε και επιπλέον περιορισμό

xi,j=0

οπότε να αποφύγουμε αναθέσεις συνεργατών που έχουν χαρακτηριστεί ως «εξαιρετικά ανεπιθύμητοι» (π.χ. THRESHOLD = -1.7). Αν έχουμε πολλούς γύρους, εδώ μπορεί να οδηγηθούμε σε αδυναμία λύσης (τσακωμένοι όλοι με όλους!), οπότε τότε προσπαθούμε ξανά μειώνοντας το THRESHOLD.