Task Allocation in Networks

Francis Bloch[1] Lilian Hartmann[2] Luca Paolo Merlino[3]
Dotan Persitz[4]

[1]Paris School of Economics and Université Paris 1 Panthéon Sorbonne

[2]PreDoc Student, Harvard

[3]Université Libre de Bruxelles

[4]Coller School of Management, Tel Aviv University

February 2024, Technion

Partially Overlapping Teams

Large research institutions and technologically innovative firms frequently use several research teams, **partially overlapping in their expertise**, to complete a given set of tasks.

Example: The European Organization for Nuclear Research (CERN)

- The scientists in CERN, operator of the largest particle physics laboratory in the world, are organized in nine large research teams (“Experiments”) that are partially overlapping in the expertise of their members and in their explicit goals.
- Di Stefano and Micheli (2022) study the interaction between two of the experiments and write:

... despite sharing institutional linkages (through CERN), using the same key resource (LHC), and having their headquarters physically co-located (in Geneva, Switzerland), each organization has a strong incentive to be the first to make any discovery to secure recognition, research funds, and human resources. (page 6)

The Problem of Partially Overlapping Teams

- In such organizations it seems plausible that while **the institutional goal** is to complete all tasks as quickly as possible, **the teams' goal** is to collect as many credits on projects' completions as possible.
- We study this setting as an optimal dynamic many-to-one matching problem.

General Setting

- Consider a planner that is required to complete a given set of tasks using a given set of agents.
- The main restriction is that each agent (experiment) is capable of handling only a subset of the tasks with which the planner is confronted (suitability network or bipartite graph).
- Each agent can engage in one task at a time.
- Once an agent engages with a task, the completion time is stochastic.
- In the centralized approach, the planner optimally assigns agents to tasks so that the total completion time is as short as possible.
- In the decentralized approach, the agents assign themselves into tasks, so that each agent tries to maximize the number of tasks she completed.

Literature I

- Matching (Huge literature on static matching):
  - Bipartite graphs: Echenique et al. (2013) and Demuynck and Salman (2022).
  - Dynamic matching when markets change over time: Doval (2022), Baccara et al. (2020, 2023) and many others recently.
  - Repeated matching (Exploration-Exploitation): Jeloudar et al. (2021).
- Search:
  - Weitzman (1979) and Gittins (1979) showed that the optimal sequence depends on a single index - simple strategy.
  - Following work showed that in many cases the optimal search is hard (E.g. parallel search (Vishwanath (1988, 1992)), outside options (Doval (2018)) and complementarities (Eliaz et al. (2022))).
- Team Formation (We have to study it better)
  - Optimal contracts when externalities between workers exist: Bernstein and Winter (2012) and Dasaratha et al. (2023).
  - Generalists and specialists: Hart and Moore (2005).

Literature II

- Networks:
  - Buyer-Seller Networks: Kranton and Minehart (2001) and Corominas-Bosch (2004).
- Scheduling:
  - Our problem is a stochastic scheduling problem with machine eligibility restrictions.
  - The central planner minimizes makespan.
  - The workers' game is not studied (as far as we know).
  - This literature assumes that a job can be processed by (at most) one machine at a time.
  - If (i) No machine eligibility restrictions and (ii) Heterogeneous jobs, then an optimizing planner applies a Longest Expected Processing Time First policy (Pinedo (2022)).
  - M_j is the set of machines that are eligible to process job j. If (i) For every j and k: $M_j \cap M_k \in \{\emptyset, M_j, M_k\}$ and (ii) Heterogeneous machines, then an optimizing planner assigns the Least Flexible Job to the Fastest Machine (Pinedo and Reed (2013)).

The Constraints Graph

- A finite set of agents, N, indexed by $i \in \{1, 2..., n\}$.
- A finite set of tasks, T, indexed by $k \in \{1, 2..., m\}$.
- Agent i can perform the set of tasks $T_i \subseteq T$ (cardinality: t_i).
- Let $E = \{(i, k) | k \in T_i\}$.
- These defines a bipartite graph $G =< N, T, E >$ (we assume it is connected).
- N_k is the set of agents who can complete task k (cardinality: n_k).

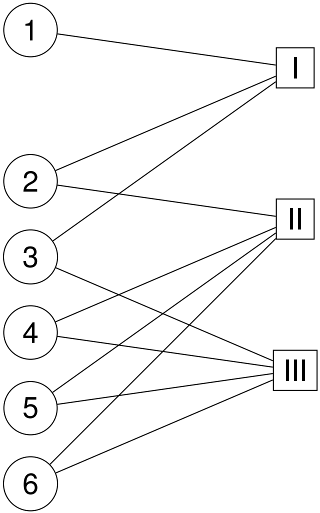
Task Completion

- Time is continuous.
- Every agent working on a task has a stream of successes that follow a Poisson distribution with fixed parameter λ.
- Therefore, if a subset of agents $L \subset N$ with cardinality l works on the same task k, the stream of successes follows a Poisson distribution with parameter $l\lambda$.
- Therefore the time until first completion (which is the relevant random variable here) follows an exponential distribution with parameter $l\lambda$ (the mean is $\frac{1}{l\lambda}$).
- Agents can be reassigned at any point in time.
- The process ends when all m tasks have been completed successfully.

Objectives

- The centralized version of the model:
  - The planner assigns agents to tasks.
  - The objective of the planner is to complete all tasks as quickly as possible (in expectancy).
- The decentralized version of the model:
  - Agents assign themselves to tasks.
  - The objective of each agent is to complete successfully as many tasks as possible (in expectancy).
  - Markov Perfect Equilibrium.

The Constraints' Graph ($\lambda = 1$)



The Centralized Problem

- Tasks II and III are symmetric so the planner is indifferent regarding the assignment of agents 4, 5 and 6.
- To which task should Agents 2 and 3 be assigned?
- Their assignment affects the tasks for the final two periods.
- Due to symmetry we focus on Agent 2.

The Centralized Problem: Agent 2 to Task I

- Assignment to task I increases the chance to be left with tasks II and III.
- Eliminating Task I first guarantees:
  - 5 active agents in the second period.
  - 4 active agents in the third period.

The Centralized Problem: Agent 2 to Task II

- Assignment to task II increases the chance to be left with tasks I and III.
- Eliminating Task II first guarantees:
  - 6 active agents in the second period.
  - But what about the third period? In period 2:
    - Agents 1 and 2 are assigned to Task I.
    - Agents 4, 5 and 6 are assigned to Task III.
    - Agent 3 can be assigned to both.
    - She is assigned to Task I to increase the probability that Task III will be the final task (4 active agents for the final task are better than 3).
    - Hence, in the final period: In probability $\frac{1}{2}$, 4 agents are active and in probability $\frac{1}{2}$, 3 agents are active.

The Centralized Problem: Summary

- It turns out that eliminating Task I first, makes the completion time shorter.
  - A path of 5 and 4 leads to an expected completion time of $\frac{9}{20}$.
  - A probability of $\frac{1}{2}$ for a path of 6 and 4 and a probability of $\frac{1}{2}$ for a path of 6 and 3 leads to an expected completion time of $\frac{11}{24} > \frac{9}{20}$.
- Therefore, Agents 2 and 3 should be assigned to Task I first.
- Summarizing the optimal solution for the planner:
  - Begin by assigning Agents 1, 2 and 3 to Task *I* and indifferent regarding the assignment of Agents 4, 5 and 6.
  - If Task *I* is completed first, the planner is indifferent regarding future assignments.
  - If Task *II* (*III*) is completed first, Agent 3 (2) is assigned to Task I.

The Centralized Problem: Note

- Note that Task I can be completed by only 3 agents, while Tasks II and III can be solved by 4 agents.
- Therefore, we say that Task I is harder than Tasks II and III.
- In the example, the planner tends to assign agents to the hardest task they can complete.
- In fact, it is easy to show that two additional “1” agents turn the assignments away from Task I.

The Decentralized Problem

- Tasks II and III are symmetric so agents 4, 5 and 6 are indifferent.
- To which task should Agents 2 and 3 assign themselves?
- Their assignment affects the tasks for the final two periods.
- Due to symmetry we focus on Agent 2.

The Decentralized Problem: Agent 2 to Task I

- Assignment to task I increases the chance to be left with tasks II and III.
- Eliminating Task I first guarantees:
  - 5 competitors in the second period.
  - In the third period, probability $\frac{1}{2}$ for 4 competitors and probability $\frac{1}{2}$ for elimination.

The Decentralized Problem: Agent 2 to Task II

- Assignment to task II increases the chance to be left with tasks I and III.
- Eliminating Task II first guarantees:
  - 6 active agents in the second period.
  - But what about the third period? In period 2:
    - Agents 1 and 2 are assigned to Task I.
    - Agents 4, 5 and 6 are assigned to Task III.
    - Agent 3 can assign herself to both.
    - She assigns herself to Task III to increase the probability that Task I will be the final task (weaker competition).
    - Hence, in the final period: In probability $\frac{1}{3}$, Agent 2 is eliminated and in probability $\frac{2}{3}$, Agents 1, 2 and 3 compete for the final task.

The Decentralized Problem: Summary

- It turns out that eliminating Task II first, leads to higher expected number of completions.
  - The expected number of completions when Task I is eliminated first is $\frac{1}{5} + \frac{1}{2} \times \frac{1}{4} = \frac{13}{40}$
  - The expected number of completions when Task II is eliminated first is $\frac{1}{6} + \frac{2}{3} \times \frac{1}{3} = \frac{7}{18} > \frac{13}{40}$.
- Therefore, Agent 2 should assign herself to Task II first.
- Summarizing the unique Markov Perfect Equilibrium (up to indifference):
  - Agent 1 assigns herself to Task *I*, Agent 2 assigns herself to Task *II* and Agent 3 assigns herself to Task *III*. Agents 4, 5 and 6 are indifferent between Tasks *II* and *III*.
  - If Task *I* is completed first, all agents are either indifferent or with no choice to make.
  - If Task *II* (*III*) is completed first, Agent 3 (2) assigns herself to Task *III* (*II*).

The Decentralized Problem: Note

- In the example, agents tend to engage first with the easiest task they can complete.
- Two additional “1” agents turn the self assignments towards Task I.

General Simple Lemma

Definition

Task k is harder than Task $\bar{k}$ if the number of agents that are capable of working on Task k is smaller than the number of agents that are capable of working on Task $\bar{k}$.

Lemma

Suppose that there are only two uncompleted tasks: A and B.

- *If $n_A < n_B$, the planner assigns agents who can work on both tasks to the "harder" task A, whereas, in equilibrium, any agent who can work on both tasks, chooses to work on the "easier" task B.*
- *If $n_A = n_B$, both the planner and the agents are indifferent among the possible assignments of the agents who can work on both tasks.*

Intuition

- In both the centralized and decentralized cases: The decision in period 1 may (or may not) affect the number of active agents only in period 2.
    1. The allocation of the planner does not affect the time it takes for the first of the two tasks to be completed.
    2. The choice of the agents does not affect the competition intensity in the current stage.
- Therefore, in the centralized case the planner wishes to assign those that can work on both tasks to the harder task.
- In the decentralized case, agents that can work on both tasks would choose the easier one to increase the probability of weaker competition in period 2.
- Can we generalize to more than two tasks?

Notation

Let $G =< N, T, E >$ be a bipartite graph.

- Denote by $\mathcal{N} = \{N_1, ..., N_m\}$ the set that includes, for each task in T, the set of all agents that can engage with it ($N_k = \{i \in N | (i, k) \in E\}$).
- Let $A, B \in T$ be two tasks. Denote by $\mathcal{N}_{A,B} = \mathcal{N} \setminus \{N_A, N_B\}$, the set that includes, for each task in T, excluding A and B, the set of all agents that can engage with it.
- Let $k \in T$ and denote by $\mathcal{N}^k = \{N_l \in \mathcal{N} | n_l > n_k\}$ the set that includes, for each task in T that can be engaged by more agents than Task k ("easier than k"), the set of all agents that can engage with it.
- $\mathcal{N}^k_{A,B}$.

Union Size Invariance

Definition

G satisfies the **Union Size Invariance** property if for any two tasks A and B:

1. If $n_A = n_B$ then for any possible union C of sets in $\mathcal{N}_{A,B}$, $|N_A \cup C| = |N_B \cup C|$.
2. If $n_A < n_B$ then for any possible union C of sets in $\mathcal{N}_{A,B}$, $|N_A \cup C| \leq |N_B \cup C|$.

In every subgraph that includes task A and does not include task B there are less workers than in the same subgraph where task A is replaced by task B.

Strong Union Size Invariance

Order the tasks from the hardest to the easiest.

Definition

G satisfies the **Strong Union Size Invariance** property if it satisfies:

1. Union Size Invariance.
2. Submodularity: For any two consecutive tasks, j and k, in the hard-easy ordering and for any possible union C of sets in $\mathcal{N}^k$: $|N_k \cup C| - |C| \geq |N_j \cup N_k \cup C| - |N_k \cup C|$.
3. Increasing Differences: for any three tasks k, A and B such that $n_k < n_A < n_B$ and for any possible union C of sets in $\mathcal{N}^k_{A,B}$: $|N_k \cup N_A \cup C| - |N_A \cup C| \leq |N_k \cup N_B \cup C| - |N_B \cup C|$.

We currently have no good interpretation for the additional conditions.

Centralized Problem

The Planner's Problem

If $G =< N, T, E >$ satisfies Union Size Invariance, then a planner that acts optimally sends, in each period, each agent who is linked to multiple tasks to work on the task which is linked to the smallest number of agents (the "hardest" task).

Centralized Problem: Insights

Main insights:

- The assignment of agents to tasks determines the probability distribution on the graphs in the next period.
- Therefore, to minimize the expected time of completion, the planner wishes to choose an assignment that attaches high probability to bipartite graphs that lead to short expected time of completion starting from the next period.
- Calculating the optimal expected completion time for each subgraph with more than two tasks is generally very complicated.
- If the bipartite graph satisfies Union Size Invariance the planner can ignore this complexity since the optimal expected completion time for each subgraph is captured by the number of agents that are capable of working on each task.

Centralized Problem: Union Size Invariance

- For each subset of size k of tasks, let $(n_1, \ldots, n_k)$ the vector of the number of agents that are capable to complete each task.
- Denote by D^k the set of all such vectors.
- Union Size Invariance guarantees that for every k there exists a unique function $g_k : D^k \to \mathbb{N}$ such that $g_k(n_1, \ldots, n_k) = |N_1 \cup \cdots \cup N_k|$ and that g_k is symmetric and increasing.
- That is, the size of the union is determined by the "size" of the tasks and independently of the exact subgraph structure.
- As a result, an optimal assignment of the planner is to assign agents to the "hardest" task possible (increasing the probability for larger unions in the following periods).

Decentralized Problem

The Agents' Problem

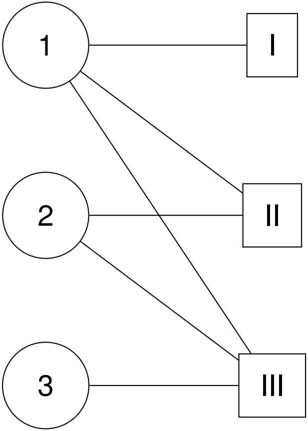
If $G =< N, T, E >$ satisfies Strong Union Size Invariance, the unique Markov perfect equilibrium of the agents' game, is such that agents who are linked to multiple tasks always work on the task which is linked to the largest number of agents (the "easiest" task).

Decentralized Problem: Insights

Main insights:

- In every period, the probability of the agent to get rewarded depends on the total number of active agents.
- Therefore, the choice of the agent in period t has no effect on her chances to complete a task in period t.
- Thus, the agent chooses the task that maximizes the number of completions over all the subgraphs that can emerge from completing tasks she can engage with.
- The problem, again, is that calculating the optimal expected number of completions for each subgraph with more than two tasks is generally very complicated.
- An additional complication here stems from the fact that the agents care only about the tasks they can complete.
- The technical part of the proof shows that Strong Union Size Invariance guarantees that choosing the “easiest” task is the unique MPE of the decentralized game.

The “Ranked tasks” bipartite graphs



The “Ranked tasks” bipartite graphs: Characterization

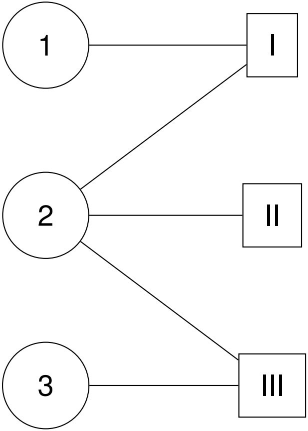
- Bipartite Chain Networks.
- Each Agent i is characterized by an integer $q_i \in \{1, 2, \ldots, m\}$ such that $T_i = \{q_i, q_i + 1, \ldots, m\}$.
- For example, tasks are ranked by:
  - Objective difficulty (Task 1 is the hardest).
  - Security clearance (Task 1 is the most confidential).
- The “Ranked tasks” bipartite graph satisfies Strong Union Size Invariance.

Corollary

If G is the “Ranked tasks” bipartite graph:

- *An optimizing planner assigns, in each period, each agent to the “hardest” uncompleted task she can work on.*
- *The unique Markov perfect equilibrium is such that tasks are being completed from the “easiest” to the “hardest” (“Low Hanging Fruits”).*

The "Specialists-Generalists" bipartite graphs



The "Specialists-Generalists" bipartite graphs: Characterization

- Two types of agents. Specialists can only work on one task (T_i is a singleton containing an element from T), and generalists who can work on all tasks ($T_i = T$).
- The "Specialists-Generalists" bipartite graph satisfies Strong Union Size Invariance.

Corollary

If G is the "Specialist-Generalists" bipartite graph:

- *An optimizing planner assigns, in each period, the Generalists to the uncompleted task with the smallest number of Specialists.*
- *The unique Markov perfect equilibrium is such that all Generalists work on the uncompleted task with the highest number of Specialists.*

The Case of Three Tasks

- Suppose that every agent that can complete exactly two of the tasks can complete tasks 1 and 2. If $n_1 > n_2 > n_3$ then an optimal planner assigns agents to the “hardest” available task.
- Comments:
  1. There is an example that satisfies this requirement but does not satisfy Union Size Invariance.
  2. When this requirement is not satisfied there are examples where an optimal planner does not assign agents to the “hardest” tasks available.
  3. This characterization can be generalized to any number of tasks (conjecture).
  4. There is an example that satisfies this requirement but “solving the easiest task first” does not constitute an MPE in the workers’ game
  5. An additional requirement is needed for the workers’ game to have a “solving the easiest task first” MPE. Still, an example exists such that Union Size Invariance is not satisfied.

Complexity

- Mathematically, the Planner's Problem is a Stationary Finite Horizon Markov Decision Problem.
- Papadimitriou and Tsitsiklis (1986, 1987) show that the complexity of solving the decision problem is polynomial in the cardinality of the set of states.
- The number of states is 2^m.
- We conjecture that the Planner's Problem is NP-hard in the number of tasks.
- Reasons to doubt the conjecture are examples given in the literature on scheduling that show that in a stochastic setting sometimes relatively simpler algorithms emerge (Pinedo (2022)).

When do these approaches coincide?

- In many cases, if the agents freely select their tasks, the payoff to the planner may be very low.
- For every $l \in \{1, \ldots, m-1\}$, let $S(\mathcal{N}, l)$ be the set of subsets of $\mathcal{N}$ of size l.
- Let $F_l : S(\mathcal{N}, l) \rightarrow \mathbb{N}$ be the family of functions that assign elements of $S(\mathcal{N}, l)$ with the cardinality of their intersection (e.g. $F_3(\{N_1, N_2, N_3\}) = |N_1 \cap N_2 \cap N_3|$).
- $G =< N, T, E >$ is a Strongly Symmetric Bipartite Graph if for every $l \in \{1, \ldots, m-1\}$ the function F_l is constant.

Coincidence

Let $G =< N, T, E >$. The optimal policy of the planner and the unique MPE of the agents' game coincide if and only if G is a Strongly Symmetric Bipartite Graph.

Price of Anarchy

- We compute the expected completion time of the project by the planner $EC^p(n_1, ..., n_m)$ and in the workers' game $EC^w(n_1, ..., n_m)$.
- We define the price of anarchy of suitability graph G as

$$PA(G) = \max_{n_1,\dots,n_m} \lim_{n\to\infty} \frac{EC^w(n_1, ..., n_m)}{EC^p(n_1, ..., n_m)}.$$

- Hard to compute generally.
- If $m = 2$, $PA(G) \approx 1.207$.
- If G is a ranked tasks model, $PA(G) \leq m - 1 + \mathbb{O}(m)$.

Bridging the Gap

- Suppose that the planner cannot assign workers to tasks. Instead she can assign rewards to completed tasks (ignore any budget constraint of the planner).
- If the rewards are contingent on the actual G (that is, determined every period) the planner can always assign rewards that will induce the agents to choose her preferred task.
- We say that a task is exclusive if it can be completed by a single worker.
- If the rewards are not contingent we can show that a reward scheme that implements the planner's optimal policy exists if and only if there is no exclusive task.

Future directions

- Necessary and sufficient conditions for the behaviors we discussed.
- Characterize cases where preemptive strategies (i.e. involve reassignment before completion) are optimal. Is it connected to the previous point?
- Allow for heterogeneity in the productivity of workers and tasks.
- Endogenize the workers' choice of effort.
- Network formation (for CERN it is actually team formation).

Thanks


THANK
YOU
GRACIAS
ARIGATO
SHUKURIA
DANKSCHEEN
TASHAKKUR ATU
YAQHANYELAY
SUKSAMA
EKHMET
GRAZIE
MEHRBANI
BOLZÏN
MERCI
BİYAN
SHUKRIA
TINGKI
GOZAIMASHITA
EFCHARISTO
JUSPAXAR
KOMAPSUMNIDA
MAAKE
PALDIES
SPASSIBO
SNACHALHUYA
NUHUN
CHALTU
WABEEJA
MAITEKA
YUSPAGARĀTAM
HUI
SPASIBO
DENKAUJA
NENACHALHYA
UNALCHEESH
GUI
HATUR
EKOJU
SIKOMO
MAKETAI
MINMONCHAR
MERASTAWHY
GAEJTHO
AGUYJE
FAKAAUE
TAVTAPUCH
MEDAWAGSE
LAH
ATTO
SANCO