Title
Object Salience in the Division of Labor: Experimental Evidence

Journal
Management Science, 62(7)

ISSN
0025-1909

Authors
Raveendran, Marlo
Puranam, Phanish
Warglien, Massimo

Publication Date
2016-07-01

DOI
10.1287/mnsc.2015.2216

Peer reviewed

DIVISION OF LABOR THROUGH SELF-SELECTION

Marlo Raveendran
marlo.raveendran@ucr.edu

Phanish Puranam
phanish.puranam@insead.edu

Massimo Warglien
warglien@unive.it

Forthcoming in Organization Science

ABSTRACT

Self-selection based division of labor has gained visibility through its role in varied organizational contexts such as non-hierarchical firms, agile teams, and project-based organizations. Yet we know relatively little about the precise conditions under which it can outperform the traditional allocation of work to workers by managers. We develop a computational agent-based model that conceives of division of labor as a matching process between workers' skills and tasks. This allows us to examine in detail when and why different approaches to division of labor may enjoy a relative advantage. We find a specific confluence of conditions under which self-selection has an advantage over traditional staffing practices arising from matching: when employees are very skilled but at only a narrow range of tasks, the task structure is decomposable and employee availability is unforeseeable. Absent these conditions, self-selection must rely on the benefits of enhanced motivation or better matching based on worker's private information about skills, to dominate more traditional allocation processes. These boundary conditions are noteworthy both for those who study as well as those who wish to implement forms of organizing based on self-selection.

In formal organizations, the division of labor is a centralized process in which managers exercise the right to design tasks, as well as the right to assign tasks to workers (Simon, 1951). The traditional version of this process is embodied in staffing practices which follow a typical sequence: analyze the task structure, design positions according to job analysis, make a job evaluation and assess the availability of current employees (or post a call to get external applicants); then select the most fit individual as soon as possible (Baron and Kreps, 1999, Chapter 14). In broad terms, this process is characterized by the attempt by managers to match the best available individual to vacant tasks as soon as possible.

Yet, a notable trend in today's business world is to allow individuals to self-select into their tasks. There are an increasing number of prominent instances in which the principle of self-selection by individuals has replaced traditional staffing processes as the basis for division of labor within a firm (Puranam, Alexy, and Reitzig, 2014). Self-organizing teams (Laloux, 2014), less-hierarchical firms (Lee and Edmondson, 2017), and holacracies (Bernstein, Bunch, Canner, and Lee, 2016; Robertson, 2015), all incorporate this element, in addition to well-known instances outside firms such as open source software development (Shah, 2006; von Hippel and von Krogh, 2003) and problem solving contests (Jeppesen and Lakhani, 2010).

In principle, division of labor through self-selection depicts individuals who select tasks (based on their skills) that in their understanding contribute to the overall goals of the team or organization. The scope of application of self-selection based division of labor within firms can vary, ranging from the entire workforce (e.g., all teams at the Dutch nursing services provider Buurtzorg, or at the U.S. based video game developer Valve; Laloux, 2014; Puranam and Håkonsson, 2015) to organizing particular project teams (e.g., in global management consultancies such as McKinsey or BCG, or in "agile" software development teams). For example, in Buurtzorg's nursing services organization, teams of 10-15 nurses self-select tasks within their district. The team manages and conducts all tasks, from providing at-home care to hiring, administration, scheduling, and training, and each nurse can choose which portfolio of activities to take on (Laloux, 2014). As a result, tasks are created and "crafted" (Wrzesniewski and Dutton, 2001) by the individual team members and task definition and scope can differ across teams in different locations. At Valve or the French auto-component maker FAVI, self-selection occurs at two levels – into particular project teams as well as into particular tasks within a team (Bernstein et al., 2016; Laloux, 2014; Puranam and Håkonsson, 2015). Despite these variations, what is common across these instances is the existence of self-selection into tasks by employees based on their own perceptions of best fit (Lee and Edmondson, 2017; also see Robertson, 2015 on holacracy).

However, we believe it is hardly time to ring the curtain down on traditional staffing processes in organizational hierarchies, in which managers with the authority to do so decide how to allocate work among employees. Even scholars who are intrigued by self-organizing processes as alternatives to hierarchical structures are nonetheless careful to note that the latter are still the dominant form in the economy today and continue to flourish even in innovation intensive sectors (Freeland and Zuckerman 2018; Lee and Edmondson, 2017; Puranam, Alexy and Reitzig, 2014). In this research, we theorize about the conditions under which self-selection would outperform traditional staffing processes as a basis for division of labor.

When managers allocate work to workers, a degree of sacrifice by workers of discretion regarding task selection is presumed (Simon, 1951). This sacrifice is compensated through extrinsic motivators such as cash, status, power, and promotion opportunities. It follows that if individuals can gain intrinsic motivation, such as greater task enjoyment, fulfilling use needs, or achieving recognition and reputation through self-selection into tasks (Lee and Edmondson, 2017; von Krogh and von Hippel, 2003), then the need for these extrinsic motivators should decline. Thus, one benefit of self-selection could simply be greater motivation. Similarly, the observability of skills should surely matter, since in many situations the worker knows her own skills better than any manager can observe (e.g., Salop and Salop, 1976; Spence, 1973). In such cases, self-selection should produce a better match between employees and the work they do, as long as employees are incentivized to select work they are competent at (Haas, Criscuolo, George, 2015; Rullani and Haefliger, 2013).

Motivation and observability of skill are intuitive considerations at the individual level that help explain the advantages of self-selection. However, as we argue, division of labor is essentially a matching process between workers and the work they do. Which matches occur and their resulting value should therefore depend not only on these *individual* attributes, but also on the *relational* attributes of workers and tasks with respect to each other. Factors such as the distribution of skills among workers (e.g., von Krogh, Spaeth, and Lakhani, 2003), the interdependence between the tasks they select (Baldwin and Clark, 2006), as well as the constraints on the matching process in terms of simultaneous or serial availability of workers and work to be matched (e.g., Cohen, March, and Olsen, 1972) should thus play a significant role in understanding the conditions under which self-selection is beneficial (Baldwin, 2015; Zenger, 2015).

The complexity involved in how these factors interact to shape the allocation process is considerable, pointing to the limits of verbal theorizing. Thus, while it is obvious that autonomous individual choices of tasks may enhance motivation and exploit superior private information about own skills for employees, how (and when) these are offset by the advantages of decision makers who can take an organization level view of possible matches between available work and available workers, under varying conditions of decomposability, specialization regimes and availability of work and workers, is less obvious.

We build a computational agent-based model, to examine under what conditions different approaches to division of labor may enjoy a relative advantage. We compare a procedure in which employees freely pick the tasks they are best skilled at (a stylized representation of self-selection) with one in which each vacant task is filled with the best-skilled available employee (a stylized representation of traditional staffing policies). Given the same set of employees and tasks, we compare how the two procedures differ in terms of aggregate task performance, task completion, and match quality. In our analysis, we hold observability and motivational effects constant by assuming that the productivity of an employee in a task is easily observable and does not depend on the allocation regime.

We find that letting employees pick the tasks they are most skilled at is advantageous in regimes involving staffing for growth (i.e. all tasks are available but employees become available at unforeseen times - as is typical in project-based organizations), with strong specialization (i.e. where most employees are very skilled at a few tasks each) and low interdependence (i.e. where each task contributes independently to overall performance). If these conditions do not hold, self-selection can only be advantageous through motivational effects, by overcoming observability challenges, or both. Note that while staffing by vacancy filling and task self-selection are usually associated with different governance modes (such as authority vs. decentralized self-organization), in our model we are comparing paradigmatic task assignment procedures and not governance modes (for example, one might find examples of self-selection within a hierarchical governance system, and centralized traditional allocation among non-hierarchical collectives).

As we show by comparison to an ideal benchmark that features optimal allocation under complete information, both procedures noted above suffer from important coordination problems. Workers allowed to pick their own tasks may suffer from inter-personal coordination failure as each worker selects his most preferred task myopically; this leaves some tasks unallocated and others overstaffed. In contrast, the procedure that fills vacant tasks with the best available employee suffers from a form of inter-temporal coordination failure as it may end up blocking better future matches by irreversibly matching the available tasks and employees today. Surprisingly, these coordination problems result in different performance consequences across the two allocation procedures, creating conditions under which one can outperform the other. While our exercise is purely theoretical, we show that the baseline results appear to have face validity when considering some of the exemplar organizations that use self-selection as a basis for division of labor (e.g., Laloux, 2014; Lee and Edmondson, 2017).

We also consider modifications to the procedures that mitigate their respective coordination failures. By allowing managers to defer allocation or to allocate employees to tasks where their added value is highest (including possibly to already staffed tasks), traditional staffing processes that follow the norm of only filling vacant tasks with best available employee could be improved upon. Conversely, developing norms that encourage employees to pick tasks where they can make the biggest difference or to avoid crowded tasks can improve on self-selection processes that simply let employees pick what they are most skilled at.

We conclude that the enthusiasm for self-managed and non-hierarchical forms of organizing that emphasize self-selection must be tempered by a consideration of our results. Our results may also indicate areas that currently do not use self-selection but could gain from doing so. The contribution of our analysis is to offer a formal conceptualization of division of labor as a matching process, and to identify a trade-off between inter-personal vs. inter-temporal coordination failures. The latter helps understand the conditions under which self-selection may prove superior to more traditional allocation processes and suggests ways to improve both processes. We also suggest directions for future theoretical development as well as possible refinements to practice involving the trade-offs between different approaches to task allocation.

The rest of this paper is organized as follows: we first review prior literature on self-selection and its individual-level and relational attributes and explain our view of the division of labor in terms of a matching problem with unique features. We then describe our model and report our key result about the tradeoff between inter-personal and inter-temporal coordination failures. We then examine modifications to the basic processes that may mitigate their respective coordination failures. Finally, we discuss three contingencies (skill observability, task interdependence, and the size of the talent pool) to study the impact of key contextual features that should affect division of labor. Our analyses give rise to a number of findings amenable to future empirical tests, possibly through (field and lab) experiments. We conclude with a discussion of our results and implications for future research.

DIVISION OF LABOR THROUGH SELF SELECTION: PRIOR LITERATURE

Self-selection is a central feature of various forms of non-hierarchical organizing both within firms (Bernstein et al., 2016; Laloux, 2014; Puranam and Håkonsson, 2015) and outside firms (Shah, 2006; von Hippel and von Krogh, 2003). Under self-selection, by definition, contributors select for themselves what tasks (or bundles of tasks constituting a "role" or a "job") to perform, rather than being assigned to their tasks or job by hierarchical processes (Baldwin and Clark, 2006; Kogut and Metiu, 2001).

Individual-Level Attributes: Motivation and observability of skills

Researchers have noted that self-selection has the obvious benefit of enhanced motivation. For example, some of the reasons for contribution through self-selection in online communities include satisfying a quest for learning, gaining recognition and visibility, fulfilling use-needs, and personal enjoyment (Lakhani and Wolf, 2005; Lerner and Tirole, 2002). Contributors' choices in a self-selection regime reflect their own needs, skills, and preferences in terms of where to contribute (Wasko and Faraj, 2000). Since an individual's skills are not always easily observed by others (e.g., Salop and Salop, 1976; Spence, 1973), managerial allocation of workers to tasks may not produce an accurate match. This gives self-selection an advantage when the worker presumably knows his or her own skill better than an external observer would (e.g., Haas *et al.*, 2015; Rullani and Haefliger, 2013).[1]

Higher job motivation and better match between individual skills and tasks have also been observed when self-selection occurs during the course of job crafting—the process through which individuals alter the task, relational, and cognitive boundaries of their jobs (Wrzesniewski and Dutton, 2001). Job crafters alter their tasks – whether formally or informally – thus incorporating an element of self-selection into their sphere of responsibilities (Berg, Wrzesniewski, and Dutton, 2010). Self-selecting into crafted tasks enables job crafters to contribute to their organization in ways that their formal job does not anticipate, and simultaneously enables job crafters themselves to learn new skills or apply skills they own but rarely get to exercise (e.g., Berg, Grant, and Johnson, 2010; Berg, Wrzesniewski, and Dutton, 2010). As a result, job crafting has been positively linked with increased job satisfaction (a sense of personal fulfillment derived from the job), job effectiveness (the person's ability to fulfill the goals and expectations of her job), organizational commitment (the person's psychological attachment to the organization), work engagement (a positive state of mind while performing the job), and an enhanced sense of self-worth (Ghitulescu, 2007; Bakker, Tims, and Derks, 2012; Wrzesniewski, LoBuglio, Dutton, and Berg, 2013).

While the match of skill to tasks is also important in traditional staffing processes, the match between workers' intrinsic enjoyment of a task and their allocated task may not be particularly high. The payment of salary is in part precisely a compensation for this (e.g., Simon, 1951). Further, managers may not be able to observe worker skills as well as workers themselves do. As a consequence, two important benefits of self-selection that arise from individual level attributes are a higher level of motivation and greater alignment between skills and tasks than what would be obtained under authority based allocation (Lakhani and von Hippel, 2003; Laloux, 2014; Lee and Edmondson, 2017).

---

[1] If workers' preferences for tasks and their skills at those tasks are not aligned, this advantage would of course diminish. For instance, a tendency towards "hobbyism" may cause individuals to take on tasks they enjoy, not necessarily the ones they are competent at, diminishing the ability of self-selection to produce effective matches between skill and tasks. We have explored these factors in additional analyses and the results are available from the authors upon request. Intuitively, a high divergence between preferences and skills hurts performance in self-selection.

Relational Attributes: Specialization and Interdependence

The benefits from enhanced motivation as well as superior self-assessment of skill (relative to a third-party allocator, such as a manager) are both instances of individual characteristics that give self-selection an advantage over traditional staffing processes. However, division of labor is a matching process, that matches individuals to tasks (or bundles of tasks combined into a role or a job). This suggests that the relational attributes of both workers and tasks (and not only their individual attributes) should also play an important role in determining when self-selection enjoys an advantage. However, while there are hints of what such factors might be in prior literature, particularly based in the open source software development context, we do not yet have a definitive analysis that is more generally applicable beyond this context.

For instance, specialization is both an antecedent and a consequence of the division of labor (Smith, 1776) and has both individual-level and relational aspects. As an individual-level attribute it refers to the attainment of higher skill on some tasks by some workers, mainly through focus and repetition (Becker, 1962). If we consider an individual's skills across multiple tasks, increasing specialization implies an increase in skill at a few tasks at the expense of most others. Thus, highly specialized individuals tend to be skilled at fewer tasks while generalists tend to have more moderate skills at a greater number of tasks (Teodoridis, 2018).

Researchers studying open source communities have speculated that self-selection is aided by high levels of specialization in skills among workers. Specialization may enable entry into the community by letting individuals make specific focused contributions (e.g., von Krogh *et al.*, 2003; Wasko and Faraj, 2000). This assumes that workers are specialized before self-selecting into tasks. Alternatively, workers might start out with equal skills for all tasks but different preferences. Task-selection would, in this case, be driven by preferences, and specialization would develop endogenously via learning-by-doing.

Further, specialization also has important relational attributes. Unless every individual is uniquely highly skilled at a distinct task, a consideration of the relative skills of individuals across a set of tasks should also play an important role in allocation of tasks to individuals. Such a consideration may not arise naturally in self-selection, because it typically ignores information about the suitability of other workers for the task that a worker selects. When worker X selects task 1, it is because for X their own skills are best suited to task 1. Due to self-interest or myopia, X does not consider the possibility that another worker may in fact be better suited to undertake task 1 than X. This can occur because of non-simultaneous entry – as in project initiation at Valve which consciously mimics open source processes (Baldwin, 2015; Zenger, 2015). However, even when all employees are simultaneously present (as when a team decided on how to self-allocate tasks among themselves e.g., Raveendran, Puranam, and Warglien, 2016), there is a significant collective action problem: optimal matching of tasks and employees requires coordination such that some employees end up with sub-optimal (for them) tasks in order to maximize overall skill values. This problem can be the result either of imperfect alignment of interests, of inability to communicate information about relative expertise, or both. The evidence on teams is quite conclusive that such problems are very common indeed (e.g., expertise recognition, Argote and Fahrenkopf, 2016; Argote and Ren, 2012; Littlepage, Robison, and Reddington, 1997; Littlepage and Silbiger, 1992). It may therefore be useful to understand the relational implications of specialization beyond the individual, specifically how the relative skill distributions of individuals may affect self-selection for a given regime of specialization.

Second, the nature of linkages between tasks is potentially an important relational attribute that should shape the efficacy of matching. In the context of open-source software communities, relatively low levels of interdependence between tasks (i.e. task structure decomposability) has been argued to allow for parallel and distributed (i.e. non physically collocated) work (Kogut and Metiu, 2001: 258) as well as attract contributions because of the possibility of exchange and re-use of work among contributors (Baldwin and Clark, 2006: 1116). However, these effects may well be idiosyncratic to open-source software development, as more generally self-selection need not involve either distributed work or exchange/recombination of contributions; for example, Buurtzorg's nursing delivery system relies on self-selection for team organization yet team members are collocated and the core tasks – in-home patient visits – cannot be recombined.

Further, in the case of open source software development, the founders do *not* lay out a fully specified task structure as a menu from which subsequent entrants choose tasks. Instead, the very act of selecting what to do may specify the task division, just as the slices of a cake become defined as individuals cut themselves portions. Thus, tasks can remain latent and undefined until they are instantiated through the interest of a contributor with the requisite skills and motives to contribute (Lakhani and Panetta, 2006). How individuals self-select some tasks therefore may also shape the interdependence between those and remaining clusters of tasks. However, the bundling of elementary tasks may be independent from the self-selection of the job to execute: In a more general setting, we could imagine a separation between task division – which may be authority based, and task allocation – which can occur through self-selection.

In sum, we know that division of labor through self-selection appears to differ from traditional allocation of workers to tasks in terms of the freedom to independently choose tasks that are deemed suitable for self (vs. having them allocated by another individual such as a manager), and the consequent benefits to motivation and observability of skills that arise. On the flipside, employees may be less coordinated in self-selection, compared to traditional staffing processes in which the authority to make decisions about allocating individuals to tasks based on organization level considerations is invested in managers. To understand the implications of these differences, we first develop the idea of division of labor as a matching process. This then sets the stage for the analysis of the conditions under which self-selection, despite being less coordinated, may nevertheless outperform traditional staffing processes.

DIVISION OF LABOR AS A MATCHING PROCESS WITH UNIQUE ATTRIBUTES

While it is intuitive to consider procedures for conducting division of labor as types of matching processes between tasks and workers, there are also significant differences between them as well as from matching problems in general. An extensive literature on matching exists in economics and operations research, starting from the seminal contribution of Gale and Shapley (1962). Algorithms have been developed in economics to solve matching problems, often grounded in rigorous mathematical analysis (for a review, see Niederle, Roth, and Sonmez, 2008). In operations research, there is also a tradition of analyzing sequential matching problems that began with Derman, Lieberman and Ross (1972) and Albright (1974) (also see Bearden et al., 2005; Chun and Sumichrast, 2006). There are three unique features of division of labor within organizations that make the insights of these prior matching models a useful reference point rather than a complete solution: multi-dimensional skills, serial entry with limited information, and switching costs.[2]

First, employee skills are multi-dimensional (Becker, 1962). Each employee can be skilled at multiple tasks, but to varying degrees (e.g., Teodoridis, 2018; Teodoridis, Bikard, and Vakili, 2018). The distribution of employee's skills across tasks can vary across different regimes of specialization. For instance, individual workers will differ in their skills across tasks (intra-agent specialization), and workers will also differ in the tasks for which they are best-skilled (inter-agent specialization). In the limit, if each agent is maximally skilled at a single task that no other agent is maximally skilled at, then matching between tasks and agents would be trivial under almost any procedure. However, in the more general case of varying distributions of skills across tasks for agents (i.e. different regimes of specialization), the nature of these distributions is likely to be a critical parameter in the process of division of labor (Mintzberg, 1979; Smith, 1776). Prior matching models do not accommodate the comparative study of different allocation procedures under different regimes of specialization, conceptualized as varying skill distributions over multiple tasks.

Second, the serial and unforeseeable entry of employees and tasks into the system makes the problem different from the matching processes typically modeled, where both sides of the matching process are simultaneously present, or if arriving sequentially, they do so with a known arrival distribution. In practice, it is often the case that either tasks become available for allocation in an unforeseeable sequence (e.g., a basic HR process in most large corporations involves staffing newly vacated or created positions), employees "come-off" other projects and become available to work on new projects in an unforeseeable sequence (e.g., in project-based software and R&D organizations), or both. As we will elaborate below, non-simultaneous and unforeseeable arrival of tasks and employees has different and surprising performance implications across different task allocation processes.

[2] An interesting parallel literature models division of labor in social insects (see Beshers and Fewell, 2001 for a review). Here, division of labor results from autonomous decisions made by each worker to perform a task. Workers are assumed to be adaptive rather than foresighted. Models try to accommodate for changing availability of tasks, heterogeneous predispositions to tasks (often represented by task-specific individual thresholds of activation), inhibition effects of others' choices, and decentralized communication.

Third, in the case of division labor, switching costs are significant and matches cannot easily be unmade. In Adam Smith's (1776) original discussion, three benefits of the division of labor in the pin factory were described: the improved productivity of the worker, the saving in time lost in switching tasks, and the development of new methods of working (including mechanization) arising from specialization. Mintzberg noted that at the root of all three benefits is repetition (1979: 70); in particular, repetition of a task cluster that requires similar inputs of skill and efforts, which consequently entails narrow cognitive scope, and allows rapid amortization of fixed costs. Put differently, switching can entail significant opportunity costs, lessening the advantages of division of labor, and may therefore not be feasible. In addition to these efficiency-based arguments, organizations likely take motivational consequences of task switching into account (e.g., the effect on employees of being replaced by better performing colleagues).

In the next section, we describe an agent-based model of division of labor as a matching process that is sensitive to these issues.

MODEL DESCRIPTION

In our model, we compare two archetypical arrangements for division of labor: Process "A" is a stylized representation of what one may observe in a traditional staffing process. A structure of tasks (we use this synonymously with jobs or roles, for our purposes) exists and is typically the result of formal design efforts around task structure and role specification. The consequences of different ways of defining the structure of tasks is expressed indirectly in our model in terms of attributes such as interdependencies between tasks and the resulting distribution of skills for tasks across employees. As tasks become available, either through new job creation or turnover, the allocator (e.g., the HR department) assesses the available labor pool internally and externally (e.g., by posting advertisements and reviewing candidates). The allocator's objective is to fill the vacancy as soon as possible picking the best available candidate (i.e. the candidate with the highest skill for the task) from among available workers (e.g., Chadwick and Dabu, 2009; Ethiraj and Garg, 2012). Process A can therefore be characterized as one in which the allocator is aiming to "*fill every vacancy with the best available person.*"

Process "B" is a stylized representation of self-selection, in which employees are free to choose from the set of jobs or tasks based on personal preference. We assume that employees' skills align with their preferences, i.e. that employees prefer tasks for which their skill value is high (we also explore alternatives later). If the tasks have been predefined, employees select their preferred task among all the tasks (irrespective of others' choices); if the task structure is ill-defined, employees create their own task and thus create the emergent task structure though their choices. The consequences of different ways of "carving up" the structure of tasks is expressed indirectly in our model in terms of attributes such as interdependencies between tasks, and the resulting distribution of skills for tasks across employees. The key feature of process B is that all employees pick their tasks independently, without consideration of organization level implications or the suitability of other employees for the task that they themselves select, each allocator (employee) is therefore free to *"pick what they like."*

In our baseline analysis, all other features are kept constant between process A and B (for a summary, please see Table 1): The costs of switching are assumed to be high enough to make allocations irreversible. Further, the individual-level attributes of motivation and observability of skills are held constant: we do not assume any information asymmetry between the allocator and employees in terms of assessing skill for a task; and worker productivity is assumed to be the same for a task in either allocation process. We assume that all allocators (whether employees or managers) are able to observe which tasks have already been staffed.

INSERT TABLE 1 ABOUT HERE

Task Environment

In our model, the task environment is characterized by a set of N tasks. These tasks can be interpreted as individual tasks or clusters of tasks bundled together into jobs or roles – we will refer to them as *tasks* for brevity but the conclusions apply equally to settings where these tasks capture jobs or roles. Tasks are chosen by or allocated to M employees. In the baseline setting we also assume that there is low interdependence between tasks. (Please refer to the Technical Appendix for information on technical details of the model).

Timing of Task and Employee availability. An important source of variation in the task environment is the timing of the availability of tasks and employees. A possible situation is one where all tasks in a project, as well as all employees available to work on it, are visible and can be *simultaneously* compared by an allocator in order to find matches. The initiation of a new project with a given set of employees is an instance of such a situation (it is equivalent to costless reshuffling of employees to tasks whenever a better match arises - something that is in practice ruled out by switching costs). The polar opposite case is the one where the tasks and employees arrive in random order (for simplicity, we assume one at a time). This is equivalent to a random pairing up of tasks with employees with no consideration of skills and specialization (Cohen et al., 1972; Lomi, Conaldi, and Tonellato, 2012). This *garbage can* situation provides another benchmark for comparison. Neither is likely to be very realistic, with more typical situations involving project *growth* as employees become available at different and unforeseeable moments to staff a known set of tasks, or *replacement* situations, where employees are selected from a known pool to staff tasks that fall vacant at unforeseeable points in time.[3] These four cases are summarized in Table 2 and illustrated in Figure 1.

INSERT FIGURE 1 & TABLE 2 ABOUT HERE

Specialization regime

We examine how performance in different allocation processes differs across specialization regimes. In high specialization regimes all employees tend to be highly skilled at relatively fewer tasks; which of the N tasks each employee is best at differs across employees and is determined randomly. In low specialization regimes, employees are about equally skilled at all tasks, but the absolute level of skill at any task is lower, capturing the trade-off in skill between specialists and generalists (Becker, 1962). We model this through a skill distribution for every employee in which their skills across all tasks sum to 1, yet they have a non-zero skill value for each task. The overall shape of the distribution is the same for all employees within a regime, but across employees, the location of the peak skill differs randomly. By altering the shape of the distribution for all employees from flatter to more peaked we can model regimes of low or high specialization. By modelling specialization regimes in this way, we ensure that each employee has the same

[3] For an instance of process B, a pool of internally available talent (such as the "bench" in IT services companies) might allocate themselves to tasks as they become available (https://medium.com/some-personal-thoughts/the-bench-in-it-companies-expense-or-investment-6f7511d28176 (accessed on 03/10/2020).

degree of specialization in a given regime (intra-agent specialization is the same for each agent) while allowing variation in inter-agent specialization (in which task each agent is best at).

For instance, assume that skill values in a specialization regime are drawn from a Normal distribution, with mean = 0 and standard deviation $\sigma = 1$. We can take advantage of the reshaping of the curve with changes in the standard deviation $\sigma \in (0,1)$: As σ increases, the curve flattens which captures lower specialization (Figure 2, panel a); as σ decreases, the curve steepens which captures higher specialization (Figure 2, panel b). To generate each employee's skill values within a given specialization regime, we sample x-values from that reshaped normal distribution at a fixed interval from the mean and normalize the resulting values to sum to 1. This results in skill values denoted by s that are all quite close together for low specialization regimes ($s1, s2, s3$ in Figure 2a). In contrast, in high specialization regimes the difference between skill values across tasks for the same individual is initially large ($s1, s2, s3$ in Figure 2b). Figure 2 only illustrates how three skill values are drawn – for the model we draw $N = 50$ skill values and normalize all 50 skill values to sum to 1. Figure 2, panel (c) shows that the difference in skill values between each employee's best and worst task is very high (close to 1) for high specialization regimes, and significantly lower (under 0.4) for low specialization regimes.

INSERT FIGURE 2 ABOUT HERE

The parameter σ in the Normal distribution thus tunes the nature of a regime of specialization, rather than the skills of any particular employee: at high specialization, each employee is good at very few tasks, but not necessarily the same set of tasks; at low specialization, each employee is fairly good at a greater number of tasks. This captures the trade-off in terms of depth vs. breadth of skills within individuals, while also allowing for differences across individuals in the tasks they are best skilled at. Analogous to the Normal distribution, we can generate different specialization regimes using the Dirichlet distribution, using the parameter α.[4]

[4] The Normal distribution provides an intuitive and familiar basis for the drawing of skill values in varying regimes. However, it does require a number of assumptions and steps (e.g., drawing probabilities for equidistant x-values, choosing the interval, normalizing) to derive the skill values for each individual. Equivalent results can be achieve through a single parameter in the Dirichlet distribution. Details on the latter can be found in the Technical Appendix. All our results hold using either distribution, the results in the paper show results from the Dirichlet distribution.

Choice process

Choice by both the allocator and worker is assumed to involve the best match with certainty. We explore later the impact of other plausible assumptions (e.g., imperfect information about workers' skills). Under process A, allocators aim to choose employees from the available pool with the highest skills for each available unoccupied task. Effectively, each individual is allocated to an unoccupied task in which their skill is highest (or randomly allocated among two tasks if her skills are identical). If all tasks and employees are simultaneously available, the optimal match can be obtained using the well-known "Hungarian" algorithm which forms the basis of a number of algorithms in network flows and matching theory (Frank, 2005; Kuhn, 1955; [5] an appendix with technical details is included in the Online Supplement). However, in all other cases (i.e. replacement, growth and garbage can), the allocator in A must try to staff available tasks with best available employees, aiming to ensure no tasks are left unstaffed. In process B, each employee selects tasks based on their own skills alone. Thus, the availability of other employees is not relevant in B. What matters for the employees in B is whether all tasks are simultaneously available to select from or not. Figure 1 provides a simple example to highlight how the choice process and the availability of tasks and workers interact.

Outcome Variables

We compare the two systems of division of labor on three metrics: organization level performance (which is increasing in skill-match between employees and tasks), matching completeness (whether tasks or employees are left unmatched – unmatched tasks and employees are indicated in red in Figure 1), and matching quality (number of tasks staffed in a way that is non-optimal for the individual i.e. they do not get their first-preference match, and average skill of matched workers). To compute organization level performance, we take the sum of the skill values across tasks of the employees allocated to those tasks. If a task is left unstaffed, it contributes nothing to organization level performance (effectively imposing an opportunity cost of unstaffed tasks which increases with task interdependence; the latter is equivalent to

[5] The Hungarian algorithm was originally developed to solve the assignment problem of jobs to workers, and subsequently employed for cost minimization (Kuhn, 1955). We adapt this algorithm to our problem of division of labor (Kamrani, Ayani, and Karimson, 2010: 42).

imposing a penalty for each incomplete task). If more than one employee chose the same task (overstaffing), we assume some effort is wasted. Thus, even though multiple employees chose that same task, only one value is entered into organization level performance. We include only the maximum skill value (the "best shot") among the employees who selected the same task in the sum of skills across all allocated tasks (Kogut and Metiu, 2001: 259).[6]

RESULTS

Baseline Comparison between Task Allocation Processes A and B

For all results, we compute the model for 1,000 iterations and present average results to eliminate any artefacts of random sampling (of task and employee entry order). In the baseline analyses, we set the number of tasks equal to number of employees, $N = M = 50$. Further, we assume independence between tasks. The baseline results thus look purely at task allocation differences in the two processes when the task division is identical, and the underlying task structure is fully decomposable. This analysis is useful to understand the key mechanism in the model, which we subsequently examine with more complex settings to understand the boundary conditions. In the analyses, we track changes in organization performance, matching completeness, and matching quality across all four cases of task and employee availability as we vary the specialization regime.

To determine how A and B perform relative to each other, it is not necessary to compute the model for all situations of task and employee availability (see Table 3 for a summary). In the simultaneous allocation situation (Case I), it is obvious that A will outperform B: Employees in B disregard the skills and choices of other employees, whereas A aims for completeness of matching (i.e. avoids understaffing and aims to staff each available task with the best available employee). Since all employees and tasks are simultaneously available, we can assume that the allocator can apply the Hungarian algorithm (also denoted by "H"), which produces the optimal match. The computations required for this algorithm increase rapidly

[6] The "best shot" approach constitutes an unfavorable assumption for B, as the skills of others are simply ignored- one could make the case that a number of high skilled employees working on the same task will improve the output. Hence, by relaxing this assumption, B performance will increase relative to A. On the other hand if one assumes that overstaffing results in a reduction of skill for the task (such as taking the *average* of all allocated employees), performance for B will decrease.

(to the cubic power) with the number of parameters (number of tasks and employees), so this conclusion does depend on computability constraints. Nevertheless, we can say that complete staffing using the Hungarian algorithm in A will outperform simultaneous allocation via B in Case I.

INSERT TABLE 3 ABOUT HERE

To determine the relative performance of A and B in cases II and IV, computation is equally unnecessary. In Case IV (*garbage can*) both A and B are equivalent to a random pairing between tasks and employees as there is no information on either the entry of tasks or workers. Their performance must therefore be identical. In Case II (*replacement*) there is information about workers but not on the arrival of tasks, so A cannot invoke the Hungarian algorithm. Nonetheless, as tasks become available, A can pick the best available worker. In contast, the assumptions we make about B in the baseline analysis ensure that it will perform poorly relative to A because of a "pile-up" problem, as all employees will take on the first available task since they do not coordinate with each other. This is perhaps why, empirically, we do not seem to observe the use of pure self-selection (B) in situations involving staffing for replacement. This ability of the model to consider this unobserved counterfactual and reveal why we do not observe it is useful in its own right.

The interesting and ambiguous case is that of staffing for *growth* (Case III), where projects grow as employees become available to staff them. Here information on all tasks is available, but employees enter in unforeseeable order, one at a time. This prevents A from applying the Hungarian algorithm but does allow matching of each arriving worker with the best available task given the worker's skill values. On the other hand, B does not suffer a pile-up problem as all tasks are visible simultaneously for the sequentially arriving workers to pick from. Here, it is hard to say whether A or B will dominate without actually computing the model, though it is noteworthy that almost every empirical instance of self-selection we observe lies in this quadrant.

The results shown in Figure 3 offer some insight into why this might be the case. Panel (a) compares organization performance of B(*growth*) and A(*growth*) (taking the performance difference, $B - A$) on the y-axis with increasing specialization on the x-axis. ***We find that A outperforms B for low and medium levels of specialization, while B outperforms A under regimes of high specialization***. This is despite the fact that A always outperforms B in terms of match completeness (see Figure 3b): no tasks are ever left undone in A (100% of tasks are single-matches, i.e. one-worker-one task), while in B about 36% of tasks are left unallocated across the range of specialization regimes. As a result, about one quarter of tasks is overstaffed, with an average of 2.4 employees per over-staffed task in B and a peak of 3.8 workers on the most overstaffed one, (vs. zero overstaffed tasks in A).

On the other hand, B always outperforms A in terms of match quality: In B, each employee always performs his/her best task while only half the employees are allocated to their first-preference match in A, as shown in Figure 3c. We contrast these results with the *organizationally* optimal allocation (Hungarian algorithm) as a purely theoretical benchmark as it requires information conditions unavailable in the growth case (also shown in Figure 3c). It is interesting to note that maximum organizational performance requires personally sub-optimal allocation for a sizable number of employees. Finally, Figure 3d highlights the higher match quality in B over A by examining the average allocated worker skill across specialization regimes. Here, only workers that actually contribute to organization performance are counted in B (i.e. the maximum skill among the 2.4 workers for any overstaffed task). Across the entire range of specialization regimes, the average match quality in B is higher than in A – even though by construction there is no private information in the model as skills and allocations are freely observable by all in both processes.

INSERT FIGURE 3 ABOUT HERE

To summarize, we find that process A, which models traditional staffing, outperforms process B, modeling self-selection, in allocation situations where tasks and employees are available simultaneously (Case I), as well as in staffing for replacement (Case II). A and B perform equally poorly under random allocation (Case IV), while B outperforms A in situations of staffing for growth (Case III), in regimes of high specialization and low interdependence. We discuss below what gives process B, despite it producing poorly coordinated choices (because workers do not take any other workers' choices into account) an advantage over the choices of A.

Mechanism: Inter-temporal vs. inter-personal coordination failures

The mechanism underlying the switch in performance in growth situations between A and B as specialization increases from moderate to high rests on a tradeoff between (1) blocking – which is high in

A, and zero in B; and (2) over/under-staffing – which is high in B, and zero in A. Blocking, or the opportunity cost of finding a better match for a given task in the future, is high in A because of its imperative to match one worker to every task, and leave no task unoccupied. Since the growth case presumes that employees only become available with delay, the allocator may select a task for a certain employee now, even though a better-skilled employee for that task comes along later. Thus, he effectively faces an *inter-temporal* coordination failure. B does not suffer from such an inter-temporal coordination failure, since every worker simply picks (once) the task at which they have the highest skills.

However, the opportunity cost of over/under staffing is high in B because of the purely parochial way in which workers select tasks. Each worker simply picks her highest-skilled task, regardless of other workers' selection of the same. If multiple workers pick the same task, a number of other tasks will turn out to be unstaffed. The strong organization performance under high specialization for B is therefore driven by the possibility of incomplete but superior matching between individual skill and task despite the fact that B therefore suffers from *inter-personal* coordination failure among workers.

The relative performance advantage of A over B thus hinges on this balance between complete staffing and better skill matches. As long as the skill values for the forced matches are relatively high (as they are in low and medium levels of specialization), they add up to generate the performance advantage of A over B, even in the growth case. But when skill values of forced matches fall, which is the case in high specialization regimes, B dominates.

These baseline results seem to mirror empirical observations. For example, reviews of Valve on the employment website www.glassdoor.com suggest that successful and/or safe projects end up being *over-staffed*. Further, "because teams are intended to be self-forming, it's rare that enough people will want to assume risk to all collectively embark on a new project. It's too safe and too profitable to just contribute to something that's already successful". In other words, some projects can also end up being under-staffed, leading possibly to many initiated but few completed projects, as public accounts suggest indeed has been the case at Valve (Keighley, 2020).[7] Interestingly, at Buurtzorg, the Dutch nursing organization, procedures

[7] https://arstechnica.com/gaming/2020/07/valve-secrets-spill-over-including-half-life-3-in-new-steam-documentary-app/

were explicitly designed to prevent any individual from being overburdened with too many undesirable (administrative) tasks, once it was realized that few nurses self-select into those tasks freely (Laloux, 2014). The problem of over- and under-staffing in self-selection is also well known in the open source communities. It is often the core developers who have to step in to pick up those tasks that nobody else self-selected (von Krogh et al., 2003). The fact that the model produces empirically consistent patterns in the baseline settings gives this theoretical exercise a degree of external validity and raises the plausibility of the rest of the analysis.

In sum, the differential performance between A and B in growth situations hinges on (1) the allocator's inter-temporal coordination failure: the opportunity cost of finding a better match in the future - which is high in A, and zero in B; and (2) the workers' inter-personal coordination failure: opportunity cost of over/under-staffing - which is high in B, and zero in A. Under a high specialization regime, the opportunity cost of foregone future matches in A is higher because the employees' second-best skill value is very low. As a consequence, B has the advantage over A in high specialization regimes with staffing for growth. We now exploit our understanding of this mechanism to consider modifications of both processes to alleviate the inter-temporal and inter-personal coordination problems of A and B. This also serves as a form of "mechanism test" of the model.

Mitigating the inter-temporal coordination failure in A

In the baseline analyses, we operate under the constraint that no period can pass without a match: allocators cannot defer staffing a task. This leads to inter-temporal coordination problems for the allocators, effectively reducing A's performance due to blocking. Next, we relax this assumption. In process A, we give the allocator a threshold value r for staffing: in the growth case, this would leave a given employee unallocated if her skill levels do not exceed r for any of the available tasks; and in the replacement case this would leave a given task unstaffed if none of the available employees' skill levels exceed r. Such a threshold model effectively mitigates the blocking cost that A faces in the baseline setting. In process B, we allow employees to wait for a task that matches their optimal skill value in B.

We find that allowing for deferred allocation in both A and B narrows the set of conditions under which B outperforms A in the growth case, as shown in Figure 4a. While the option to defer choices does not reduce the number of unstaffed tasks previously prevalent in B, it *does* improve the skill-to-task matching for the allocator in A. B therefore outperforms A across a smaller range of specialization regimes. The allocator can now strive for better matches first, at the cost of understaffing certain tasks. However, the deferral requires the allocator to hold accurate information regarding the arrival distribution of skills to be effective: only with an adequate threshold value will A outperform B for a wider range of specialization regimes.

INSERT FIGURE 4 ABOUT HERE

In contrast, in the replacement situation (where all employees are present but tasks become available in random order, Figure 4b), it is process B that benefits more from deferring matches. Compared to the baseline, where A outperformed B under *all* specialization values, B now outperforms A for very high specialization regimes across all threshold values. In B, deferral effectively removes the pile-up problem (but not the inter-personal coordination failure), such that each employee now waits for and selects the task with their highest respective skill value. The baseline and deferral analyses together highlight sharply the differential vulnerability of A and B to the lack of information in division of labor; B is most vulnerable to not seeing all available tasks at the same time (i.e. replacement situations) as it suffers from the "pile-up" problem, but is indifferent to the lack of information on other workers (since choice is parochial anyway by the allocators, the employees). In contrast, A is most vulnerable to not seeing all employees at the same time (i.e. growth situations) because of the possibility of better-skilled employees becoming available in the future, as well as the possibility of better tasks appearing later (i.e. replacement). Allowing for deferred matches in growth situations does not reduce B's performance per se. In contrast, allowing for deferred matches in growth and replacement situations can benefit A but this depends on accurate knowledge of an appropriate threshold for deferral. [8]

[8] If delayed staffing is inconsequential, then the threshold can be set very high, the allocator can wait until all candidates and tasks arrive and then apply the Hungarian – this cannot be improved upon. If delay is consequential and the skill distribution is unknown to the allocator, then the threshold cannot be set very high. Thus, without accurate knowledge of an adequate threshold value, the simpler and less accurate B allocation process may still outperform A.

An alternative modification of the A process is to allow for overstaffing (which in the baseline is only allowed in B), and match employees to tasks based on their "highest added value", such that a highly-skilled worker who arrives late could be allocated to an already staffed task, if the new arrivals' contribution to the occupied task is higher than any contribution she could make to unoccupied tasks. This hybrid of A and B (because it combines allocation that is mindful of other's choices with allowance for overstaffing) helps mitigate the inter-temporal coordination failure by overcoming blocking while minimizing the cost of overstaffing. Figure 4c shows that this "modified A" process outperforms both basic allocation processes A and B (performance is displayed in absolute terms in this figure). We show that it is indeed the reduction in blocking (modified A facilitates a greater number of first-preference matches which were blocked by early non-optimal matches in A, Figure 4d) and the resulting higher match quality (average skill-values of matched workers is above the baseline A, Figure 4e) that lead to the performance increase. Offsetting this is an increase in the number of unallocated tasks (across specialization regimes) from zero to a moderate 12% and an average number of workers per task from 1 in A to 1.2 in modified A, with a maximum of 2.2 workers on the most overstaffed task.

We recognize that modifying traditional staffing processes from A to modified A may have to contend with context specific constraints – such as the possibility that employees might react negatively to having a higher skilled colleague being added to their task, in effect setting the value of their own contribution to zero. It might therefore require careful piloting to assess whether the gain in allocation through such a modification sufficiently offsets any possible costs due to lowered morale.

Mitigating the inter-personal coordination failure in B

The baseline results assumed that under B employees took no account of the choices of other employees and were indifferent to working with others on the same task. Instead of myopically picking their best task, one might consider norms such that employees might be motivated to self-select the task that they would "make the biggest difference" at. Thus, if their best task was already occupied but they are relatively highly skilled at a second, unoccupied task, the employee would pick the latter. Figure 4c shows that this "modified B" process would outperform both basic allocation processes A and B (this modified B process is equivalent to the "modified A" process in its allocation and performance implications). Effectively, modified B would result in a reduction of overstaffing from an average of 26% overstaffed tasks and 2.4 workers per overstaffed task in the baseline, to an average of 12% overstaffed tasks with 1.2 workers per overstaffed task in modified B. This reduction in overstaffing, however, would come at the cost of reduced match quality: the number of first-preference matches drops from 50 (out of 50) to a range between 30 to 38.5 (from low to high specialization, Figure 4d) and the average skill of matched workers in modified B is slightly lower than in the baseline B (Figure 4e).

Alternatively, employees may simply be motivated to pick tasks that are less crowded (without any consideration of where they can make the biggest difference). If we assume that employees prefer to pick tasks that have few or no occupants, then performance in this second modification of B matches or exceeds A's performance across the entire range of specialization regimes (Figure 4f). While the positive effect of allowing employees to freely choose tasks based on their highest skill levels remains, the negative crowding preference introduces a disciplining mechanism that prevents extreme levels of crowding and reduces the number of unstaffed tasks. Effectively, employees are encouraged to look for their "second best" task-skill match if another employee already occupies their first task choice. The cost of over/under-staffing in these hybrid versions of B (which allow for overstaffing while adding a consideration of other's choices) is lower compared to the baseline case. However, we also acknowledge that creating norms for such mindful-of-others self-selection may not be easy in all contexts. Again, a careful piloting may be required to assess whether such modifications produces benefits to offset the somewhat diminished autonomy in choice relative to baseline B.

In sum, both sets of modifications to A and B mitigate their respective weaknesses – inter-temporal and inter-personal coordination failures, and bring them both closer to the optimal Hungarian algorithm (and therefore each other; these modified processes can be seen as hybrids of A and B).

Three Performance Contingencies: Observability, Interdependence, and Talent pool

We conclude our analyses by examining three contingencies that influence the relative performance advantage of A and B allocation in growth situations: observability of skills, task interdependence, and depth of talent pool. The model implementation underlying these discussions as well as the figures are included in the Online Supplement.

Why accurate observability of skills is critical for A

In the baseline model we assumed that employees' skills could be visible equally well to themselves in B as well as a third party allocator like a manager in A. One departure from such a baseline would be to introduce noise in matching for A but not for B (on the plausible assumption that it is easier for employees to know their own skills than it is for a third-party allocator because of information asymmetry). However, in such a case it is intuitive that we would create a strict disadvantage for A (we can show that B in this case outperforms A across all specialization regimes). One can also imagine scenarios in which allocators are better able to assess employee skills (through appropriate assessment and testing tools, for instance), in which case the advantage would tip towards A over B.

We can make a more subtle comparison of the two allocation processes under the assumption that both face the same levels of noise. Specifically, we examine the case when both allocator and employees may suffer from imperfect ability to observe employee skills: we continue to let allocators observe employee skills as well as employees themselves can observe their own skills, but the observations of both parties are now noisy. When noise affects the matching process, choice is assumed to involve the best match with some probability rather than with certainty.

Interestingly we find that moderate increases in noise diminish the performance of A more than that of B, particularly as specialization increases. The reason for this differential effect is that in B, even if one employee misses his best choice under noise, another employee may select that task, compensating for the initial miss. ***Overstaffing creates redundancy that compensates for noise.*** However, this option is not available in A. Since the allocator in A leaves no task unstaffed, a miss on a high-skilled task for one worker has two effects: First, the opportunity cost of a high task-to-skill-match; and second, an early mis-matched employee may now block a later high task-to-skill match. The negative externalities of these effects increase in strength with increasing specialization – higher specialization regimes have fewer high-skilled tasks per employees, which significantly increases the opportunity cost of mismatches. Thus, while overstaffing in B compensates for noise, noise in A exacerbates the blocking problem.

Why task decomposability is critical for B

We confirm that task structure decomposability, i.e. task independence, produces a strong advantage for B. The baseline analysis assumes that the underlying task structure is highly decomposable so that task interdependence across employees is negligible. With greater task interdependence (more off-diagonal "1's" in the task structure), the overall system becomes less decomposable. The possible interaction costs between tasks allocated to different employees is one obvious issue to consider as a direct cost of reduced task decomposability. However, even if we ignore interaction costs (assume it is the same for employees in A and B), we find that A already has an advantage at dealing with interdependence: lower decomposability serves to increase the opportunity costs of unallocated tasks. Given that the allocator in A leaves no tasks unallocated, A will outperform B for highly interdependent task structures, even without the advantage such a system could hold in terms of managing interaction costs.

Why a shallow talent pool favors A and a deep one favors B

The baseline model assumes that the number of tasks and employees is the same ($N = M$). Here, we explore how changes in the depth of the talent pool influence relative performance in A and B; we continue to focus on the growth case.

We find that, overall, a shallower talent pool – where there are relatively fewer employees available for a given task (i.e. $N > M$, labor shortage) – effectively increases the probability of a high skill-to-task match in A. Given that all tasks are simultaneously available and the manager in A allocates the highest skilled employee to each task, a wider selection of tasks for each employee will make a higher skill match more likely. As a result, performance in A is higher in a shallow talent pool compared to the $N = M$ setup. While a shallow talent pool in B has the same effect of making more tasks available across employees, in B in the $N = M$ model each employee already selected her highest-skilled task, so that we see little impact of a shallow talent pool on organization performance. This differential effect of shallower talent pools (or labor shortage) on A and B effectively closes the performance gap between them at higher levels of specialization, such that B only outperforms A – under shallow talent pools – for the highest level of specialization.

A deeper talent pool (or $N < M$, labor surplus), on the other hand, overcomes the under-staffing problem for B, increasing organization performance significantly above the initial results. Performance in A under a deeper talent pool remains unchanged compared to equal numbers of task and employees, since A does not suffer an understaffing problem to begin with. In contrast, while B always suffers from tasks left undone, the greater number of available employees increases the likelihood that each task is chosen by someone and reduces the percentage of tasks left unstaffed in B.

As a corollary, the negative effects of interdependence in B described above can be dampened by increasing the number of available employees. This effect is driven by the reduction in unstaffed tasks with increased depth of the talent pool.

CONCLUSION

Self-selection based division of labor is a cornerstone of several systems of non-hierarchical organizing. The best-studied of such systems so far have been online communities. Researchers have pointed to several factors that seem to be important in these contexts. Decomposability of task structures, exploited through fine grained modular architectures for instance, may create independence of action, allowing for parallel contributions (Kogut and Metiu, 2001; Lakhani and Panetta, 2007), as well as opportunities for exchange of valuable work (Baldwin and Clark, 2006). By attracting a large and diverse body of contributors, modular architectures may also stimulate and exploit specialization in skills (von Krogh et al., 2003; Wasko and Faraj, 2000) and improve the possibility of creating a close match between contributor skills and task requirements (Rullani and Haefliger, 2013; also see Haas *et al.* , 2015).

Self-selection has also gained popularity as a basis for division of labor within firms – with managerial application of the principle to holacracies, agile software development teams, and non-hierarchical organizations (Laloux, 2014; Lee and Edmondson, 2017; Puranam and Håkonsson, 2015). However, unlike the open source software context, self-selection within the firm need not involve either distributed work or exchange/recombination of contributions, and implicitly always competes with the possibility of traditional staffing by managers. To understand the conditions under which self-selection may be advantageous, we developed a computational model of division of labor as an irreversible matching process. The comparison between different matching processes reveals that the intuitions derived from the study of systems with self-selection alone may be incomplete.

Our analyses help us understand why traditional staffing processes that embody the principle of "*fill every vacancy with the best available person*" can be superior to self-selection processes where every individual "*picks what they like*" across a wide range of conditions. Unlike self-selection, which fails to coordinate choices across workers, traditional staffing explicitly takes an organization level perspective, aiming to optimize organization performance rather than myopically looking for the best match possible for each worker. Nonetheless, we discovered that there are specific conditions under which self-selection has an advantage, *even if* we held individual level attributes (such as motivation and observability of skills) constant across the allocation procedures.

We find that the traditional process outperforms self-selection when it pays to leave no task unstaffed, possibly at the cost of poor-quality matches because of blocking (i.e. matches made today are worse than what could be made later). Conversely, self-selection has an advantage when it pays to create better skill-to-task matches for individuals (i.e. every employee works on tasks they are most skilled at, their first-preference match), but at the expense of under- and over-staffed tasks. The diverse results summarized in Table 4 can all be understood with respect to this basic trade-off between inter-temporal coordination failure (leading to blocking) in traditional staffing and inter-personal coordination failure (leading to over/under staffing) in self-selection.

INSERT TABLE 4 ABOUT HERE

With increasing *interdependence* the costs of under-staffing of tasks can be dramatic, because any tasks left undone can harm the entire system. Interdependence thus creates a significant disadvantage for self-selection which is prone to leaving some tasks unstaffed.[9] With increasing *specialization*, the cost of blocking increases – because the possibilities of far superior matching in the future increase – whereas those of over-staffing declines – because the best skilled individual has very high skills. This is why self-selection

[9] We have modeled a particularly strong form of interdependence (multiplicative such that any unstaffed task reduces its interdependent tasks' values to zero), which effectively stacks the deck against self-selection because it is prone to under-staffing. Nonetheless, regimes emerge where it dominates.

gains an advantage under high specialization regimes. *Deferred allocation* reduces the cost of blocking in situations of growth tilting the scales towards traditional staffing, whereas it increases the benefit of self-selection in staffing for replacement, by preventing pile-up of employees on the first available vacant task. Norms that *avoid crowding* benefit self-selection by reducing over/under staffing. A *shallower talent pool* reduces the occurrence of blocking, because of greater task availability per employee; whereas a *deeper talent pool* reduces the occurrence of under-staffing because of more possible high-skill matches for the few available tasks. The former therefore benefits traditional allocation, the latter, self-selection. *Noise* in the task-to-employee matching process gives self-selection an advantage, because there are more opportunities for rectification through the choices of other employees – effectively an advantage of over-staffing. If the allocator gets it wrong in traditional staffing because of noise, this blocking effect cannot be rectified. Notably, all these effects would all hold *even if* there were no motivational or informational advantages to self-selection (which we know would shift the balance further in favor of self-selection).

These results also uncover some subtle aspects of both allocation procedures. Paradoxically, the effectiveness of "spontaneous coordination" seen in non-hierarchical organizations such as open source communities and self-managed teams, may actually depend on individuals' preferences for working alone on tasks. Absent such preferences, over-staffing is exacerbated in self-selection, resulting in opportunity costs. We can also infer that a valuable role for managers in traditional staffing - even if they do not undertake any dispute resolution or direction of subordinates – may simply be to prevent over- and under-staffing in a non-decomposable system to avoid the ripple effects of leaving tasks left undone.

Understanding the nature of the coordination failures affecting our archetypical task assignment procedures allows to address some of their shortcomings. Allowing managers to defer filling vacancies or allocating multiple employees to the same task (when that represents the best improvement in organization performance) can improve traditional staffing. Creating norms where employees allocate themselves to tasks where they can make the biggest difference, or at least to avoid crowded tasks can improve self-selection.

Our results have face validity when we compare them to accounts of organizations that employ self-selection practices in their operations, in that we see anecdotal evidence of over- and under-staffing as predicted (e.g., Buurtzorg, Valve). The contribution of this theoretical exercise lies in (1) uncovering important boundary conditions that would have not been easy to detect from empirical data (e.g., the availability of tasks before workers, specialization, and independence for conferring an advantage on self-selection), as well as (2) providing a deeper understanding of the mechanism underlying the relative performance differences, that go well beyond the expected motivational and informational advantages that intuitively characterize self-selection.

More generally, we think that our model may contribute to filling a relevant gap in the literature on skills and organizations. Most of the recent literature on matching in organizations has focused on firm-specific skills, and the pairing of workers with firms (e.g., Lazear, 2009). However, as Gibbons and Waldman (2004) have forcibly argued, task-specific skills are potentially more relevant for understanding the internal working of organizations. Too little is known about how the distribution and dynamics of task-specific skills affect the design and operation of organizations, and the process of division of labor – notwithstanding the original focus of Adam Smith (1776) – at this level of analysis. We provide new insights into how task-specific skill specialization affects the dynamics of matching workers to jobs, and its effect on performance under different organizational regimes. By doing so, our work brings new emphasis on a central – but under-investigated – level of analysis of organizations.

The results of our analysis offer a first window into the conditions under which each form of intra-organizational division of labor may have relative advantages. They may be seen as hypotheses to be confirmed in data. Since the counterfactual comparison between the two regimes of division of labor is unlikely to be naturally observable in the field, the need for (lab or field) experiments seems clear to progress on this agenda. Pending such exploration, we hope our results can be used to inform, if not guide managerial thinking on this matter.

REFERENCES

Albright, S. C. (1974). Optimal sequential assignments with random arrival times. ***Management Science***, *21*(1), 60-67.

Argote L, Fahrenkopf E. 2016. Knowledge transfer in organizations: The roles of members, tasks, tools, and networks. ***Organizational Behavior and Human Decision Processes***, 136: 146-159.

Argote L, Ren YQ. 2012. Transactive Memory Systems: A Microfoundation of Dynamic Capabilities. ***Journal of Management Studies***, 49: 1375-1382.

Bakker AB, Tims M, Derks D. 2012. Proactive personality and job performance: The role of job crafting and work engagement, Human Relations, 65(10): 1359-1378.

Baldwin CY, 2015. In the Shadow of the Crowd: A comment on "Valve's Way," ***Journal of Organization Design,*** 4(2): 5-7.
Baldwin CY and Clark KB 2006. The Architecture of Participation: Does Code Architecture Mitigate Free Riding in the Open Source Development Model? ***Management Science***, 52(7): 1116-1127.
Baron JN and Kreps DM. 1999. Strategic Human Resources: Frameworks for General Managers, Chapter 14, Wiley: NY.
Bearden, J. N., Murphy, R. O., & Rapoport, A. (2005). A multi-attribute extension of the secretary problem: Theory and experiments. *Journal of Mathematical Psychology*, *49*(5), 410-422.
Becker GS, 1962. Investment in Human Capital: A Theoretical Analysis. *J. Polit. Econ.* 70(5, Part 2):9–49.
Berg JM, Grant AM, Johnson V. 2010. When Callings Are Calling: Crafting Work and Leisure in Pursuit of Unanswered Occupational Callings, Organization Science, 21(5): 973-994.
Berg, J. M., Wrzesniewski, A., & Dutton, J. E. (2010). Perceiving and responding to challenges in job crafting at different ranks: When proactivity requires adaptivity. *Journal of Organizational Behavior*, *31*(2-3), 158–186.
Bernstein E, Bunch J, Canner N, Lee M. 2016. Beyond the Holacracy Hype, ***Harvard Business Review***, July-Aug.
Beshers SN, Fewell JH. 2001. Models of Division of Labor in Social Insects, ***Annual Review of Entomology***, 46: 413-440.
Chadwick C, Dabu A. 2009. Human Resources, Human Resource Management, and the Competitive Advantage of Firms: Toward a More Comprehensive Model of Causal Linkages. ***Organization Science***, 20(1):253-272.
Cohen MD, March JG, Olsen JP. 1972. A Garbage Can Model of Organizational Choice, ***Administrative Science Quarterly***, 17(1): 1-25.
Chun, Y. H., & Sumichrast, R. T. (2006). A rank-based approach to the sequential selection and assignment problem. *European Journal of Operational Research*, *174*(2), 1338-1344.
Derman, C., Lieberman, G. J., & Ross, S. M. (1972). A sequential stochastic assignment problem. *Management Science*, *18*(7), 349-355
Ethiraj SK, Garg P. 2012. The Division of Gains from Complementarities in Human-Capital-Intensive Activity. ***Organization Science,*** 23(3):725-742
Frank, A. 2005. On Kuhn's Hungarian Method—A Tribute from Hungary**, Naval Research Logistics**, 52(1): 2-5.
Freeland RF and Zuckerman EW 2018. The Problems and Promise of Hierarchy: Voice Rights and the Firm, Sociological Science, 5: 143-181.
Gale D, Shapley LS. 1962. College Admissions and the Stability of Marriage, ***American Mathematical Monthly***, 69(1): 9-15.
Ghitulescu, B. (2007). *Shaping tasks and relationships at work: Examining the antecedents and consequences of employee job crafting*: University of Pittsburgh.
Gibbons, R. and Waldman, M., 2004. Task-specific human capital. ***American Economic Review***, 94(2): 203-207.
Haanigan TR, Haans RFJ, Vakili K, Tchalian H, Glaser VL, Wang MS, Kaplan S, Jennings PD, 2019. Topic Modeling in Management Research: Rendering New Theory from Textual Data, Academy of Management Annals, 13(2): 586-632.
Haas MR, Criscuolo P, George G, 2015. Which Problems to Solve? Online knowledge sharing and attention allocation in organizations, ***Academy of Management Journal***, 58(3): 680-711/
Jeppesen LB, Lakhani KR. 2010. Marginality and Problem-Solving Effectiveness in Broadcast Search, ***Organization Science***, 21(5): 1016 – 1033.
Kamrani F, Ayani R, Karimson A. 2010. Optimizing a Business Process Model by Using Simulation, ***IEEE Workshop on Principles of Advanced and Distributed Simulation***, 2010 IEEE Xplore conference proceedings.
Kogut B, Metiu A. 2001. Open-Source Software Development and Distributed Innovation, ***Oxford Review of Economic Policy***, 17(2): 248-264.
Kuhn HW. 1995. The Hungarian method for the assignment problem, ***Naval Research Logistics Quarterly***, 2: 83–97.
Lakhani KR, Panetta JA. 2007. The principles of distributed innovation. I***nnovations*** **2**(3): 97–112.

Lakhani KR, Von Hippel E, 2003. How open source software works: “free” user-to-user assistance. ***Research Policy,*** 32(6): 923–943.

Lakhani KR, Wolf R. 2005. Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. In ***Perspectives on Free and Open Source Software,*** Feller JR *et al*. (eds). MIT Press: Cambridge, MA: 3–21.

Laloux, F. 2014. ***Reinventing Organizations***, Nelson Parker: Brussels.

Lazear, E. P. 2009. Firm-specific human capital: A skill-weights approach. ***Journal of Political Economy***, 117(5): 914-940.

Lee MY, Edmondson A. 2017. Self-Managing Organizations: Exploring the limits of less-hierarchical organizing, ***Research in Organizational Behavior***, 37: 35-58.

Lerner J, Tirole J. 2002. Some Simple Economics of Open Source. ***Journal of Industrial Economics,*** 50: 197–234.

Littlepage G, Robison W, Reddington, K. 1997. Effects of Task Experience and Group Experience on Group Performance, Member Ability, and Recognition of Expertise. ***Organizational Behavior and Human Decision Processes***, 69: 133-147.

Littlepage, G. E. and H. Silbiger, 1992. Recognition of Expertise in Decision-Making Groups:Effects of Group Size and Participation Patterns. ***Small Group Research***, 23: 344-355.

Lomi A, Conaldi G, Tonellato M. 2012. Organized Anarchies and the Network Dynamics of Decision Opportunities in an Open Source Software Project, ***In The Garbage Can Model of Organizational Choice: Looking Forward at Forty***: 363-397.

Mintzberg, H. (1979). ***The Structuring of Organizations***. Prentice-Hall: Englewood Cliffs, NJ.

Niederle M, Roth AE, Sonmez T. 2008. Matching and Market Design, ***The New Palgrave Dictionary of Economics***, Second Edition, Eds. SN Durlauf and LW Blume, Palgrave Macmillan.

Puranam D, Narayan V, and Kadiyali V. 2017. The Effect of Calorie Posting Regulation on Consumer Opinion: A Flexible Latent Dirichlet Allocation Model with Informative Priors, **M*arketing Science***, 36(5): 726-746.

Puranam P, Alexy O, Reitzig M. 2014. What’s “new” about new forms of organizing? ***Academy of Management Review***, 39(2): 162–180.

Puranam P, Håkonsson DD (2015) Valve’s way. ***Journal of Organization Design*** 4(2): 2–4.

Raveendran M, Puranam P, Warglien M. 2016. Object Salience in the Division of Labor: Experimental Evidence, ***Management Science***, 62(7): 2110-2128.

Robertson, B. (2015). *Holacracy: The New Management System for a Rapidly Changing World.* Henry Holt.

Rullani F, Haefliger S. 2013. The Periphery on Stage: The intra-organizational dynamics in online communities of creation, ***Research Policy,*** 42: 941-953.

Salop J and Salop S. 1976. Self-Selection and Turnover in the Labor Market, Quarterly Journal of Economics, 90(4): 619-627.

Shah SK. 2006. Motivation, Governance, and the Viability of Hybrid Forms. ***Management Science,*** 52(7): 1000–1014.

Simon HA. 1951. A formal theory of the employment relationship. ***Econometrica: Journal of the Econometric Society***: 293–305.

Smith, A. 1776. An Inquiry into the Nature and Causes of the Wealth of Nations. Edinburgh.

Spence M. 1973. Job Market Signaling. Quarterly Journal of Economics, 87(3): 355-374.

Teodoridis F, 2018. Understanding Team Knowledge Production: The Interrelated Roles of Technology and Expertise. *Manage. Sci.* 64(8):3625–3648.

Teodoridis F, Bikard M, Vakili K (2018) Creativity at the Knowledge Frontier: The Impact of Specialization in Fast- and Slow-paced Domains. *Adm. Sci. Q.*:000183921879338.

Von Hippel E, von Krogh G. 2003. Open source software and the “private-collective” innovation model: Issues for organization science. ***Organization Science,*** 14(2): 209–223.

Von Krogh G, Spaeth S, Lakhani KR. 2003. Community, joining, and specialization in open source software innovation: a case study. ***Research Policy***, 32(7): 1217–1241.

Wasko, M. M., S. Faraj. 2000. “It is what one does”: Why people participate and help others in electronic communities of practice. J. Strategic Inform. Systems 9(2–3) 155–173.

Wrzesniewski, A., & Dutton, J. E. (2001). Crafting a Job: Revisioning Employees as Active Crafters of Their Work. *Academy of Management Review*, *26*(2), 179–201.

Wrzesniewski A, LoBuglio N, Dutton JE, & Berg, JM. 2013. Job Crafting and Cultivating Positive Meaning and Indentity in Work, in Advances in Positive Organizational Psychology, Vol. 1: 281-302.
Zenger TR. 2015. Valve Corporation: Composing Internal Markets, ***Journal of Organization Design***, 4(2): 20-21.

TABLE 1. Baseline Assumptions about different archetypes of Division of Labor

Modeled allocation process	**Empirical phenomenon**	**Unique assumptions**	**Common assumptions**
Process A Allocator is aiming to "*fill every vacancy with the best available person*".	Traditional staffing (e.g., typical HR process)	Allocator fills every vacancy with a person, by picking best available employee for that task.	1. High switching costs prevent reshuffling. 2. No periods without allocation. 3. Arrival of task/employee is not foreseeable. 4. Skills of available employees are visible to everyone. 5. Allocations are visible to all.
Process B Each allocator is free to "*pick what they like*"	Self-selection based allocation (e.g., self-organizing teams)	Allocator picks best available task for themselves ignoring other workers' skill values.	

TABLE 2. Variations in the Timing of the Availability of Tasks and Employees

	All employees are simultaneously available	**Employees become available in unforeseeable order**
All tasks are simultaneously available	*Case I: Simultaneous* e.g., starting a new project with a team of available employees	*Case III: Growth* e.g., projects grow in staff as employees become available
Tasks become available in unforeseeable order	*Case II: Replacement* e.g., staffing a position that becomes vacant	*Case IV: "Garbage can"* e.g., projects are staffed based on random availability of tasks and employees

TABLE 3. Relative Performance of Allocation Processes A and B

	All employees are simultaneously available	**Employees become available in unforeseeable order**
All tasks are simultaneously available	*Case I: Simultaneous* A > B	*Case III: Growth* B > A under high specialization and low interdependence
Tasks become available in unforeseeable order	*Case II: Replacement* A > B	*Case IV: Garbage can* A and B perform equally

TABLE 4. Summary of Model Results and Propositions

		Division of labor: Effect of allocation based on		Predictions:	
		Process A (e.g., traditional staffing)	**Process B (e.g., self-selection)**	**B advantage over A…**	**Notes**
INDIVIDUAL-LEVEL ATTRIBUTES					
Motivation	Workers derive motivation from freedom to choose tasks.		Increases task-performance via intrinsic motivation.	… increases when worker motivation improves because of the freedom to choose task.	A by definition takes away the choice from workers.
Observability of skill	Information asymmetry, workers know own skills better than allocators do.	Worsens managers' match quality.		… increases when workers know their own skills better than managers.	Better match between worker skill and tasks in B if workers are incentivized to pick their highest-skill task.
	Noise in the task-to-employee matching process for workers and allocators.		Overstaffing rectifies any poor skill choices due to noise.	… increases with moderate noise levels because overstaffing rectifies prior poor matches.	Excessive noise is equally detrimental for A and B.
RELATIONAL ATTRIBUTES					
Interdependence	Significant interdependence between tasks, *i.e. the value of any task is contingent on other tasks being performed.*		Penalizes unstaffed task.	… decreases when tasks are highly interdependent.	A ensures no under-staffing and may also oversee interactions to ensure integration of effort.
Specialization regime	Workers are highly specialized at few tasks.	Increases the opportunity cost of blocking.	Reduces the opportunity cost of overstaffing.	… increases when workers are highly specialized.	A suffers from higher costs of blocking; B benefits from lower cost of understaffing.
Timing of choice	Deferral of allocation until multiple workers/tasks arrive.	Reduces the opportunity cost of blocking.	Prevents overstaffing in replacement.	… diminishes in growth, increases in replacement with deferral possibility.	In B, no effect on growth.
Worker crowding preferences	Workers find tasks unattractive based on what others choose.		Negative crowding reduces overstaffing.	… increases when workers prefer working alone.	In B, positive crowding worsens overstaffing.
Depth of talent pool	Variation in the number of workers available per task-match.	Shallower talent pool reduces occurrence of blocking.	Deeper talent pool reduces understaffing.	… increases when the talent pool is deeper.	In B, more workers than tasks increases high-skill matches; in A, more task options per worker improves match quality.
Maximize contribution	Allocators and workers select matches based on where the new employee's contribution is maximized.	In the presence of overstaffing this improves match quality at the cost of (some) unstaffed tasks.	Reduces overstaffing.	… equals out; the two modified (A and B) processes perform equally well and better than both basic A and B.	Implementation via staffing policy in A, via norm development or selective hiring in B.

FIGURE 1. Task allocation across four cases of task and employee availability

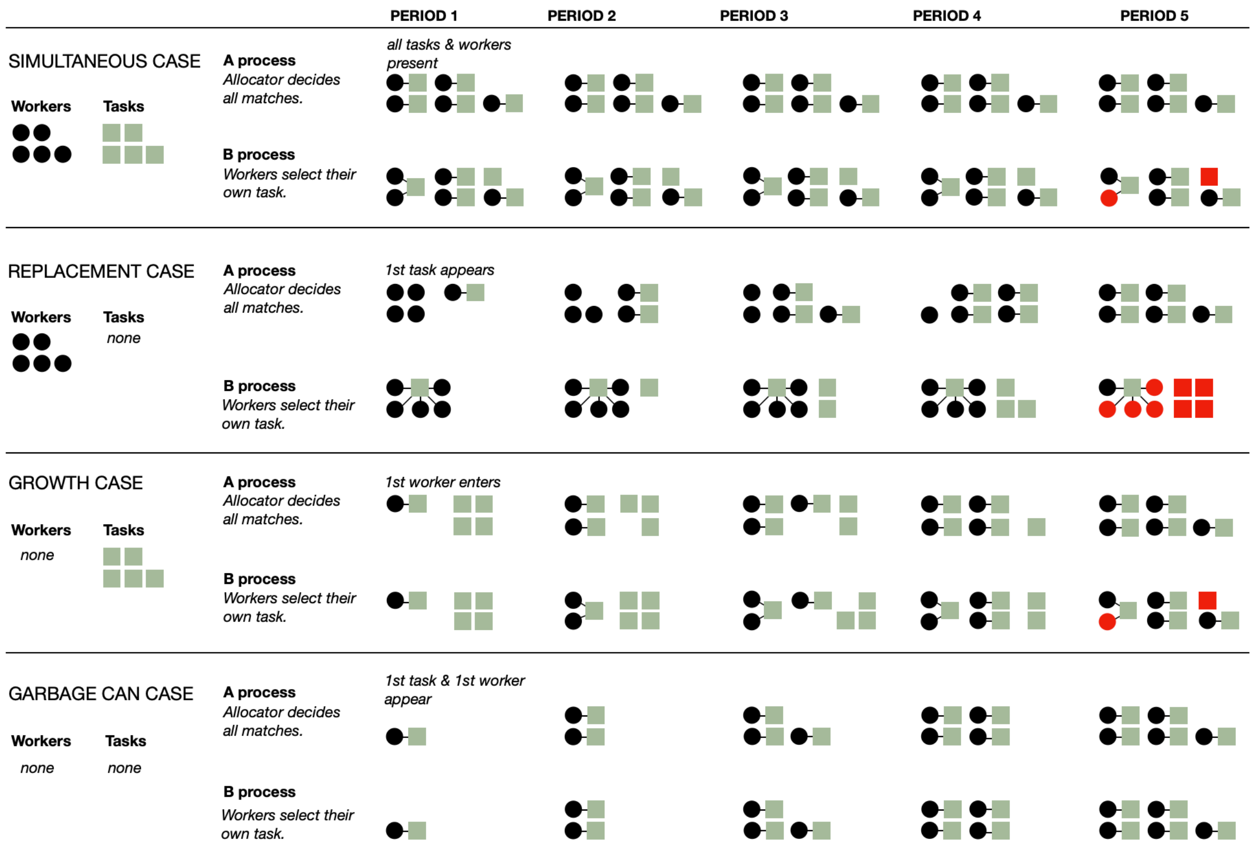


Figure 1 shows the intuition of how task and employee availability affect the different task allocation processes with five tasks and five workers. On the far left, we show what is available for each case at the start: how many workers (black circles) and how many tasks (green squares). For example, in replacement, we start with all workers and zero tasks; in garbage can, we start with no workers and no tasks. For each case, we then show period by period how task allocation or self-selection unfold. In our model, all four cases are based on the same underlying task structure; this means that the workers and the allocator make their decisions in each of these four cases based on the identical underlying skill distributions. At the end of some of these cases, some tasks are overstaffed (indicated by any red circles connected to one square) while some tasks are left unallocated (indicated by red squares).

FIGURE 2. Modeling specialization using the Normal Distribution

Panel (a) lower specialization Panel (b) higher specialization Panel (c) Intra-person skill differences

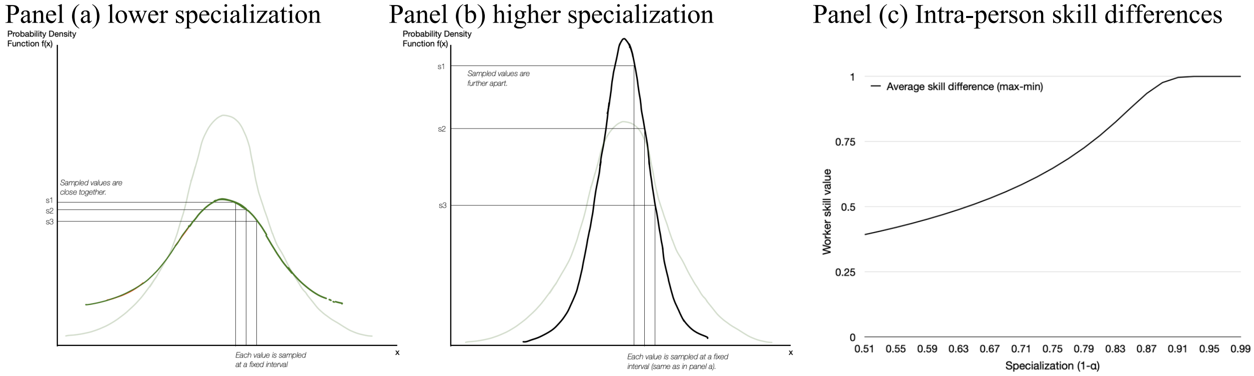
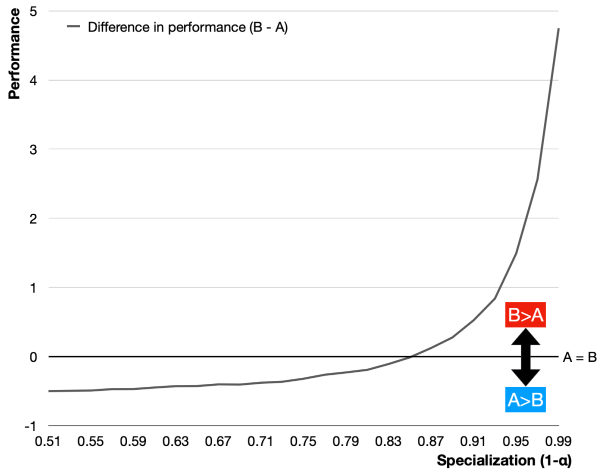
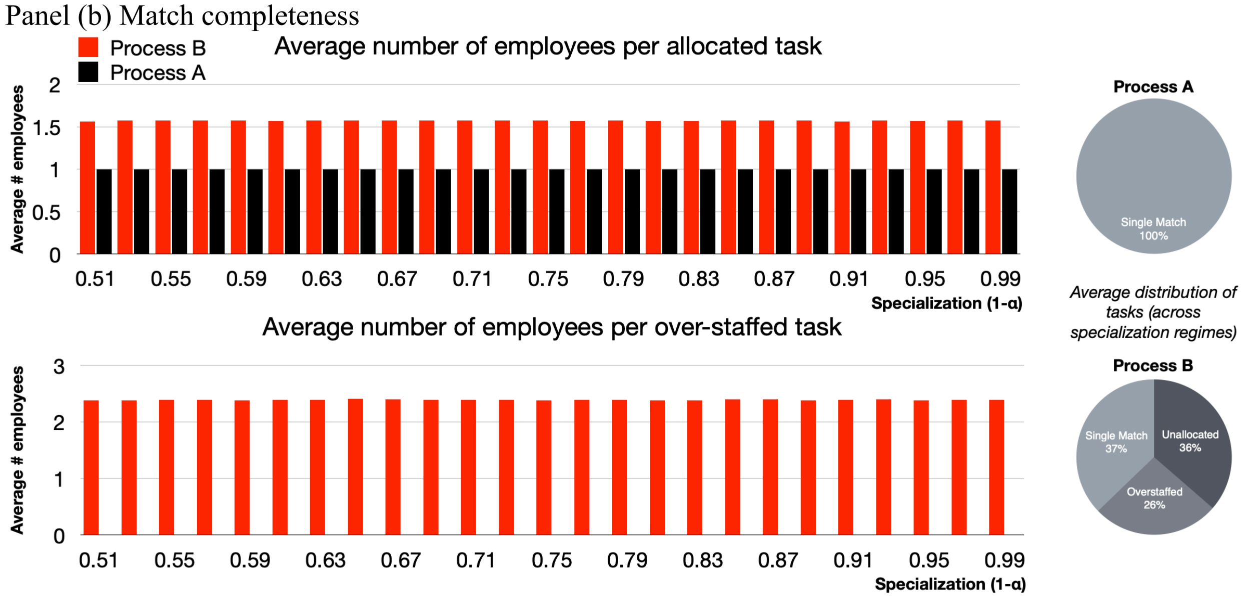


FIGURE 3 - Performance comparison: Allocation processes A and B in the baseline growth model

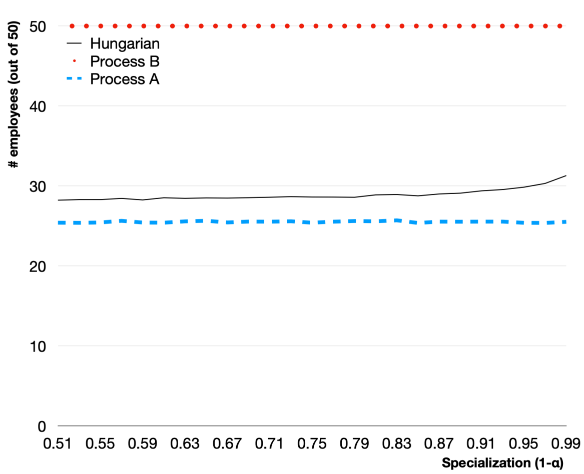
Panel (a) Performance difference (B-A)



Panel (b) Match completeness



Panel (c) First-preference matches



Panel (d) Average skill of matched worker

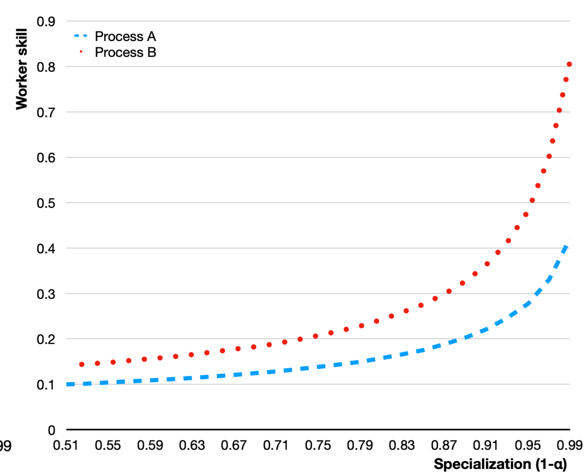
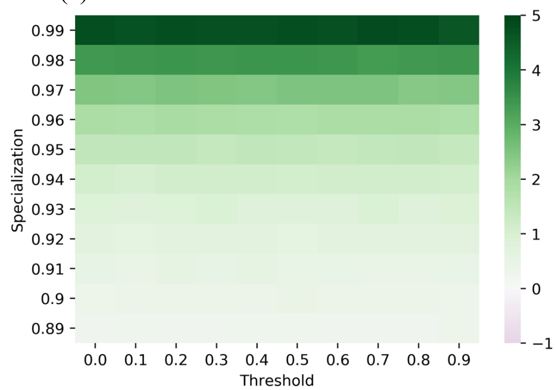
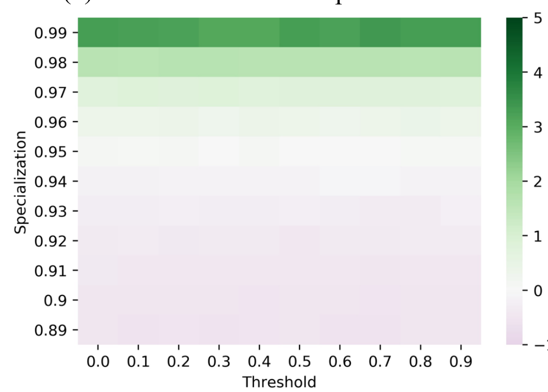


FIGURE 4 – Modified allocation processes

Panel (a) Deferred Matches: Growth

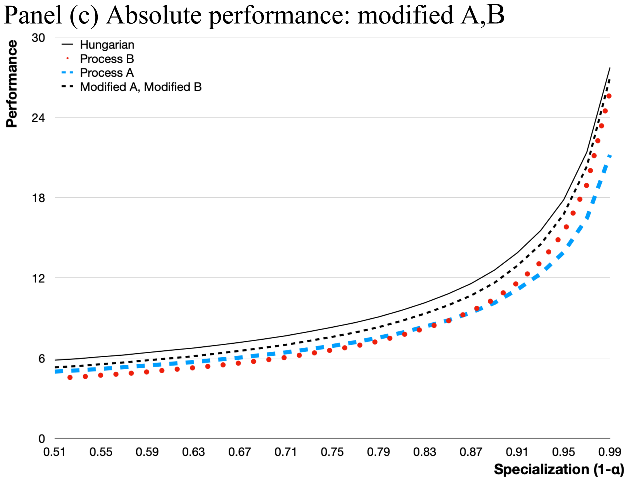


Panel (b) Deferred Matches: Replacement

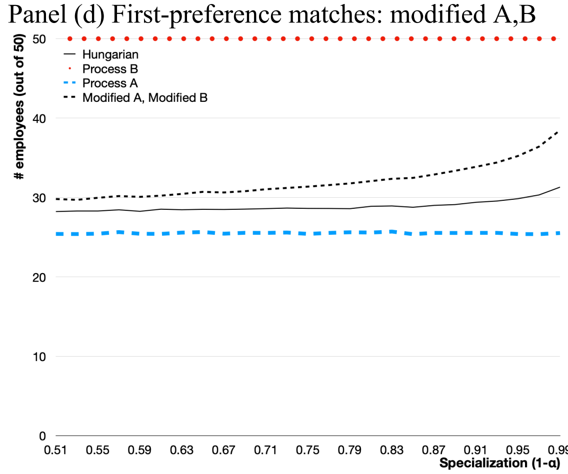


The heatmaps in panels (a) and (b) show the performance differential between B and A ($B - A$) across specialization regimes under deferred matching and various threshold values r. Dark (green) shading shows areas where B clearly outperforms A; (light) purple shading shows negative areas, where A clearly outperforms B.

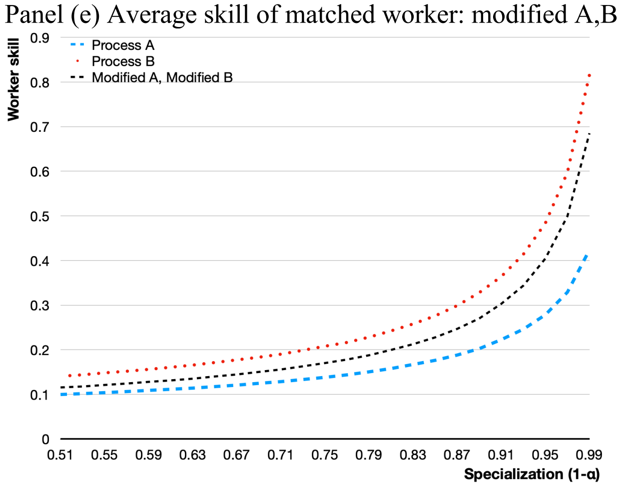
Panel (c) Absolute performance: modified A,B



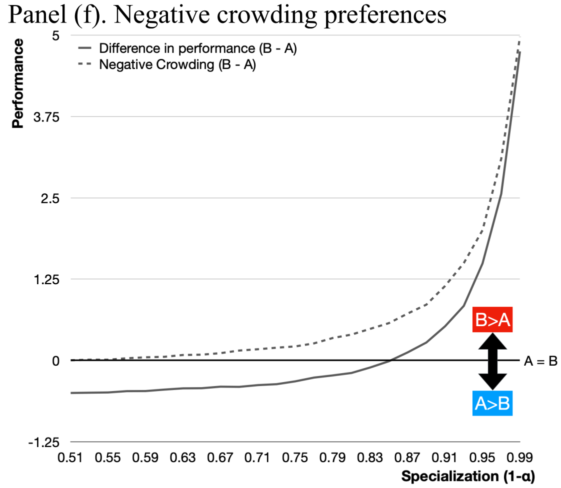
Panel (d) First-preference matches: modified A,B



Panel (e) Average skill of matched worker: modified A,B



Panel (f). Negative crowding preferences



Panels (c), (d), and (e) show the effects across performance metrics in the growth case of modified-A (***highest added value***) and modified-B (***make the biggest difference***). Panel (f) shows the effect of negative crowding for B (compared to A) in the growth case in comparison to the baseline (B-A) result.

TECHNICAL APPENDIX

Task Environment

The task environment is characterized by a set of N tasks and M employees. The *allocation* of employees to tasks at any point in time *t* is captured by the $M \times N$ asymmetric matrix L_t . If employee i is assigned to task j, then $L_{ij} = 1$, else $L_{ij} = 0$. We assume that allocators are able to observe L_t (i.e. which tasks have been staffed at time t).

Interdependence. The set of tasks that collectively contribute to the organization's performance is represented by a square matrix T of size $N \times N$. The matrix captures the patterns of interdependence between tasks: $T_{ij} = 1$ implies that task *i* is dependent on task *j* and this dependence can be unilateral. This is referred to as the *task structure* and denoted by T. In the baseline setting, tasks are assumed to have low interdependence i.e. a staffed task can have non-zero value even if other tasks are left unstaffed.[10]

Specialization regime

Employees have skills for every task. This is represented as the $M \times N$ matrix K, which gives measures of employee i's skill for task j. We explained the intuition behind the specialization regime with a Normal distribution in the main text. Here we explain how to draw the specialization regime using a symmetric Dirichlet distribution of dimension N, with the concentration parameter $\alpha \in (0, 1)$ tuning specialization. This distribution is a useful representation of probability over a set of discrete states (e.g., Haanigan et al., 2019; Puranam, Narayan and Kadiyali, 2017). We adopt it to model the distribution of skills across tasks, subject to a constraint on total skill. We use the symmetric distribution as we do not mean to impose any particular prior for any of the workers to favor any of the N tasks. While α can take on any positive value in the Dirichlet distribution, the distribution behaves differently for different ranges of concentration parameters (i.e. the range of values generated by α between 0 and 1 behaves differently from the range of values generated by α > 1). We take advantage of the properties of the distribution for $\alpha \in (0,1)$: the values of the resulting distribution tend to be less evenly distributed compared to α > 1. This is precisely what mimics the workers' tendency to be good at some, but not all, values: it is the variance of this property over the range of concentration parameter values strictly between 0 and 1 that allows us to mimic different degrees of specialization.

A given employee's skill values will always sum up to 1 across the range of N tasks. However, the specialization regime (tuned by α) changes both how skilled that employee is at any one of those tasks as well as how many tasks the employee is relatively good at. Under a high specialization regime (a low value of α) each employee has high skills for only one or two tasks and very low skills for all remaining tasks. For example, at $\alpha = 0.01$ (highly specialized) in a task environment with 50 tasks, the maximum skill value (in the range (0,1)) is 0.78, the second highest skill value is 0.16, and the skill values for the remaining 48 tasks range between 0.04 and almost zero. However, given the nature of the Dirichlet distribution, the skill values of all workers remain strictly in the open interval *between* zero and one, even for extreme specialization, and sum up to one.[11]

---

[10] Our baseline assumptions therefore stand in stark contrast to Adam Smith's setting of functional division of labor (sequential steps in the production line) where any one unallocated task would bring overall performance to zero; they are more akin to an nk landscape with zero interdependence (k=0). In additional analysis we consider cases with interdependence.

[11] We also explored the limit case of "perfect" specialization where skill values took on strictly zero or one; in this scenario, performance in B equals performance of the optimal Hungarian algorithm, since avoiding overstaffing under the latter does not add any non-zero performance benefits to the B-allocation choices. Under these conditions, A will always underperform B, unless inter-personal specialization is perfect such that each employee is highly skilled at a different, unique task, in which case A=B=H performance. We thank an anonymous reviewer for highlighting this case.

Under a low specialization regime (a high value of α) a given employee's skill values are more similar to each other across the N tasks. Given that her skill values across a particular task sum up to one, however, these skill values are all relatively lower compared to the high specialization regime. For example, at $\alpha = 0.51$ a low specialization regime, in a task environment with 50 tasks, the maximum skill value is 0.14, the second highest value is 0.10, and the remaining 48 skill values range between 0.08 and 0.00002.

Outcome Variables

To compute organization level performance, we take the sum of the skill values (across tasks) of the employees allocated to those tasks. Thus:

$$\text{Organization Level Performance } \pi = \sum_{n=1}^{N-n} \max \{s_i\} \quad (1)$$

Where N is total number of tasks;

n is the number of unallocated tasks;

$\{s_i\}$ is the set of i employees' skill values for each task.

ONLINE SUPPLEMENT

Why accurate observability of skills is critical for A

When there is noise in the matching process, choice is assumed to involve the best match with some probability rather than with certainty. This probabilistic selection is based on a behavioral rule that has robust psychological validation and is called "Luce's choice rule" or the "softmax" action selection rule (Luce, 1959; Posen and Levinthal, 2012; Sutton and Barto, 1998; see Puranam, Stieglitz, Osman, and Pillutla, 2015 for a review of the psychological evidence). In this rule, the probability of a choice depends on the strength of the skill estimate for each task.

Under B, the probability of an employee selecting a particular task i with skill s is therefore:

$$p(i) = \frac{e^{s_i/\tau}}{\sum_{i=1}^{M} e^{s_i/\tau}}$$

The parameter τ tunes the degree of noise in the choice process (the probability of the allocator choosing an alternative independent of its relative attractiveness). It can be interpreted as a slippage between intention and choice, or as a conscious decision to take actions inconsistent with current beliefs in order to explore. This formula applies equally to the growth and replacement cases.

Under A, the probability of the allocator choosing the newly available employee for task j in the growth case can be written as:

$$p(j) = \frac{e^{s_j/\tau}}{\sum_{j=1}^{N} e^{s_j/\tau}}$$

where s_j is the skill of the available employee at task j. For the replacement case, the probability of the allocator choosing the newly available task for employee m (based on her skill for that task) can similarly be written as:

$$p(m) = \frac{e^{s_m/\tau}}{\sum_{m=1}^{M} e^{s_m/\tau}}$$

where s_m is the skill of employee m at the available task. We can now compare the baseline results (which involved deterministic choice) with probabilistic selection with increasing levels of noise ($0 \leq \tau \leq 0.5$). When τ approaches zero, skills approach perfect observability, which declines as τ increases. The initial results hold in that B outperforms A under high specialization. Yet, with high levels of noise, the performance of A and B converges to a low level across all specialization regimes ($0.3 < \tau < 0.5$). Results are displayed in Figure 5a.

INSERT FIGURE 5 ABOUT HERE

Why task independence is critical for B

To examine the impact of task interdependence on relative performance in the growth case, we model interdependence using an "O-ring" performance function. For every task i in the task structure matrix T, we identify all other tasks j that the focal task is interdependent with. We then multiply the skill value of employee m_i that was allocated to task i with the average of all the skill values of employees m_j for the interdependent tasks j. If any of the interdependent tasks j was left unallocated its value is zero; in this case, task i's performance input is zero as well. Even if all tasks j are allocated, other employees' skill values will affect task i's performance contribution, so a high task-skill match for every task that task i is interdependent with becomes even more important for organization performance with increasing interdependence. This O-ring performance function is a fairly "punishing" form of interdependence. In most real-world scenarios, the absence of a complement does not bring to zero the value of an activity (because activities are unlikely to be *perfect* complements). As such, the results on interdependence presented here provide the boundary condition of interdependence in A and B.

Compared to the initial analyses (the case of no interdependence between tasks), even a moderate degree of interdependence effectively decreases total organization performance for both allocation processes, but especially for B – as shown in Figure 5b. This is because any unstaffed task j in B that is interdependent with task i effectively reduces the value of i to zero. For example, at interdependence of 0.01, an average of 25 cells in the task structure (besides the diagonal) take on the value of "1". If any of the interdependent tasks happens to be dependent on an unstaffed task, its skill value effectively becomes zero.

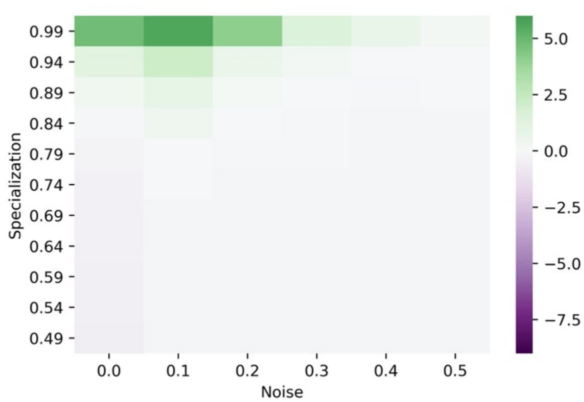
Given that an average of 18 tasks are left unstaffed in B, low degrees of interdependence are sufficient for the point at which performance of B exceeds performance of A across different specialization levels to shift upwards, so that B only outperforms A under more extreme cases of specialization. Despite the disadvantage of unstaffed tasks in B, it still outperforms A under the highest level of specialization for moderate degrees of interdependence; specifically, B outperforms A as long as the benefit of using the best task-to-employee match every time in B outweighs the benefit of match completeness in A. However, with increasing interdependence, A soon outperforms B across all specialization regimes (see Figure 5b). These results establish the soundness of common intuition that high task structure decomposability is important for B to work; they also offer an additional reason beyond the need to minimize interactions: the greater system-wide adverse consequences of tasks left undone in B.

Why a shallow talent pool favors A and a deep one favors B

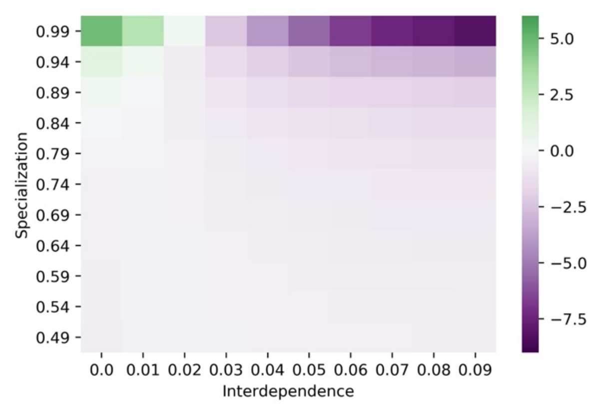
Results discussed in the paper are displayed in Figure 5c.

FIGURE 5 – Three Contingencies for Performance

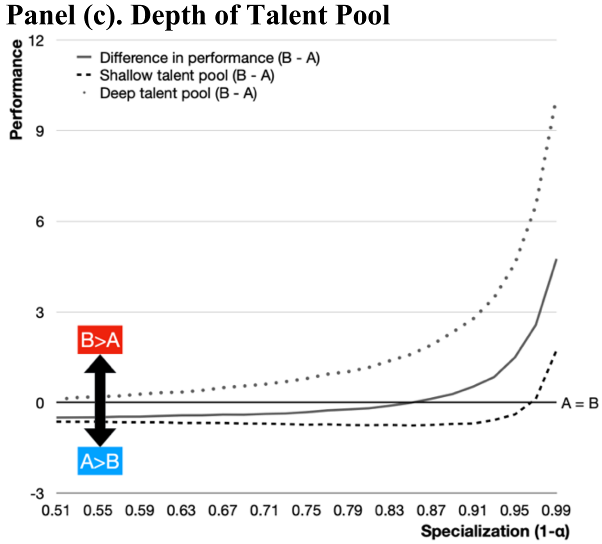
Panel (a). Observability of Skills (Growth)



Panel (b). Interdependence (Growth)



Panel (c). Depth of Talent Pool



Details on the Hungarian Algorithm

The Hungarian algorithm was originally developed to solve the assignment problem of workers to tasks, stated simply as "what is the largest number of jobs that can be assigned to qualified individuals (with no more than one job assigned to each individual)?" (Kuhn, 1955: p.84). Qualified individuals, in this context, were those that had the ability to perform a given job, where each individual may be qualified to only complete a few of the available jobs. This algorithm has become the prototype of a large number of algorithms in Combinatorial Optimization in areas such as network flows and matching theory (Frank, 2005) and has also been adapted to the context of management science (e.g., Kamrani, Ayani, and Karimson, 2010). Here, we adapt this algorithm to our problem of division of labor.

To facilitate the comparison with our model, we use the matrix interpretation of the Hungarian algorithm. Given a nonnegative $n\ x\ n$ matrix $\mathrm{A} = \alpha_{ij}$, the element α_{ij} represents the allocation of task j to agent i. This algorithm is commonly used for cost minimization problems where α_{ij} represents the cost of assigning task j to agent i. The algorithm consists of a simple sequence of steps that are applied iteratively to the matrix to identify a unique match of worker to task and task to worker, minimizing the cost for each task *and* each agent (Kuhn, 1955). We describe the steps of the algorithm below and provide a simple numerical example, given in Figure 6.

Step 1. Our objective here is to maximize (rather than minimize) the total "cost", so we start by negating all elements of the matrix.

Step 2. The Hungarian algorithm adds the value of the smallest negative element to all elements if any of the elements is negative. We therefore add 95 to all elements (in our example).

Step 3. Subtract row minima: For each *row* r_i, determine the minimum value and subtract it from each element of r_i.

Step 4. Subtract column minima: For each *column* c_i, determine the minimum value and subtract it from each element of c_i.

Step 5. Lines: Cover all zeros by drawing a minimum number of lines across rows and columns.

If the number of lines equals the size of the matrix n*, the assignment problem is solved. If the number of lines is lower than* n*, iterate between steps 5 and 6 until the number of lines equals the size of the matrix. In our example, four lines cover rows and columns in a* $\mathrm{n} = 5$ *matrix, so we continue to step 6.*

Step 6. Subtract and add the minimum value: Determine the minimum value across all *uncovered* elements and subtract that value from all *uncovered* elements; add this number to all elements that have been covered by *two* lines. In our example, we subtract 5 from all uncovered elements accordingly, and add it to those covered by two lines.

Step 7. Repeat step 5. In our example, the number of lines now equals the size of the matrix. We have solved the assignment problem, maximizing the total value with a unique match between agents and tasks.

Note that the final result is intuitive for tasks 1 and 2, while there are apparently "better" choices for tasks 3, 4, and 5. In order to see why the solution is optimal, attempt to improve the current selection for those three tasks. You will notice that an additional 4 or 7 points for those three tasks comes at a cost of -20 or more on those tasks that you are switching out.

FIGURE 6 – Numerical Example for the Hungarian Algorithm

Baseline matrix

Step 1: Negate all elements to maximize cost.

	task 1	task 2	task 3	task 4	task 5	α_{ij}		task 1	task 2	task 3	task 4	task 5
agent 1	51	75	94	54	78	**(-1)*	*agent 1*	-51	-75	-94	-54	-78
agent 2	21	3	90	48	20	**(-1)*	*agent 2*	-21	-3	-90	-48	-20
agent 3	64	95	3	62	82	**(-1)*	*agent 3*	-64	-95	-3	-62	-82
agent 4	90	77	88	46	79	**(-1)*	*agent 4*	-90	-77	-88	-46	-79
agent 5	57	69	83	55	25	**(-1)*	*agent 5*	-57	-69	-83	-55	-25

Step 2: Add the smallest value to all elements.

	task 1	task 2	task 3	task 4	task 5	α_{ij}		task 1	task 2	task 3	task 4	task 5
agent 1	-51	-75	-94	-54	-78	*(+95)*	*agent 1*	44	20	1	41	17
agent 2	-21	-3	-90	-48	-20	*(+95)*	*agent 2*	74	92	5	47	75
agent 3	-64	-95	-3	-62	-82	*(+95)*	*agent 3*	31	0	92	33	13
agent 4	-90	-77	-88	-46	-79	*(+95)*	*agent 4*	5	18	7	49	16
agent 5	-57	-69	-83	-55	-25	*(+95)*	*agent 5*	38	26	12	40	70

Step 3: Subtract row minima.

	task 1	task 2	task 3	task 4	task 5	α_{ij}		task 1	task 2	task 3	task 4	task 5
agent 1	44	20	1	41	17	*(-1)*	*agent 1*	43	19	0	40	16
agent 2	74	92	5	47	75	*(-5)*	*agent 2*	69	87	0	42	70
agent 3	31	0	92	33	13	*(-0)*	*agent 3*	31	0	92	33	13
agent 4	5	18	7	49	16	*(-5)*	*agent 4*	0	13	2	44	11
agent 5	38	26	12	40	70	*(-12)*	*agent 5*	26	14	0	28	58

Step 4: Subtract column minima.

	task 1	task 2	task 3	task 4	task 5			task 1	task 2	task 3	task 4	task 5
agent 1	43	19	0	40	16		*agent 1*	43	19	0	12	5
agent 2	69	87	0	42	70		*agent 2*	69	87	0	14	59
agent 3	31	0	92	33	13		*agent 3*	31	0	92	5	2
agent 4	0	13	2	44	11		*agent 4*	0	13	2	16	0
agent 5	26	14	0	28	58		*agent 5*	26	14	0	0	47
	(-0)	*(-0)*	*(-0)*	*(-28)*	*(-11)*	α_{ij}						

FIGURE 6 – Numerical Example for the Hungarian Algorithm (cont'd)

Step 5: Cover zeros with a minimum number of lines.

	task 1	task 2	task 3	task 4	task 5		α_{ij}
agent 1	43*	19*	0	12*	5 *		(-5)*
agent 2	69*	87*	0	14*	59*		(-5)*
agent 3	31	0	92'	5	2	x	(+5)'
agent 4	0	13	2 '	16	0	x	(+5)'
agent 5	26	14	0 '	0	47	x	(+5)'
			x				

Step 6: Subtract min value from uncovered elements.

	task 1	task 2	task 3	task 4	task 5
agent 1	38	14	0	7	0
agent 2	64	82	0	9	54
agent 3	31	0	97	5	2
agent 4	0	13	7	16	0
agent 5	26	14	5	0	47

Step 7: Cover zeros with a minimum number of lines.

	task 1	task 2	task 3	task 4	task 5	
agent 1	38	14	0	7	0	x
agent 2	64	82	0	9	54	x
agent 3	31	0	97	5	2	x
agent 4	0	13	7	16	0	x
agent 5	26	14	5	0	47	x

Solution: Optimal assignments.

	task 1	task 2	task 3	task 4	task 5
agent 1	38	14	0	7	0
agent 2	64	82	0	9	54
agent 3	31	0	97	5	2
agent 4	0	13	7	16	0
agent 5	26	14	5	0	47

Baseline Matrix: Total value of optimal allocation = 408

	task 1	task 2	task 3	task 4	task 5
agent 1	51	75	94	54	78
agent 2	21	3	90	48	20
agent 3	64	95	3	62	82
agent 4	90	77	88	46	79
agent 5	57	69	83	55	25

References

Frank, A. 2005. On Kuhn's Hungarian Method—A Tribute from Hungary**, Naval Research Logistics**, 52(1): 2-5.

Kamrani F, Ayani R, Karimson A. 2010. Optimizing a Business Process Model by Using Simulation, ***IEEE Workshop on Principles of Advanced and Distributed Simulation***, 2010 IEEE Xplore conference proceedings.

Kuhn HW. 1995. The Hungarian method for the assignment problem, ***Naval Research Logistics Quarterly***, 2: 83–97.

Luce, D. R. (1959). ***Individual choice behavior***. New York, NY: Harper and Row.

Posen HE, Levinthal DA (2012) Chasing a moving target: Exploitation and exploration in dynamic environments. ***Management Science,*** 58(3):587–601.

Puranam P, Stieglitz N, Osman M, Pillutla MM (2015) Modelling bounded rationality in organizations: Progress and prospects. ***Academy of Management Annals,*** 9(1):337–392.

Sutton, R. S., and Barto, A. G. (1998). ***Reinforcement learning: An introduction***. Cambridge: MIT Press. (BA).