How to Enable Multiple Skill Learning in a SLA Constrained Service System?

Abstract

In a knowledge based service system like IT services, a resource needs to work on service requests that require single or multiple skills. In either case, the resource is expected to learn the required skills very quickly and become productive. Due to high attrition rate and demand, service workers are given basic class room training and then rest of the training is carried out on-job. When a service worker learns multiple skills simultaneously, learning slows down due to factors like forgetting and interference. At the same time, the organization needs to meet service level agreements (SLA). We have developed a self-guiding work allocation mechanism that is designed to achieve the desired goals of upskilling and SLA success. To get realistic service times, we have modeled learning, forgetting and interference in service time estimation. Accurate estimation of service time taken by a service worker to resolve the service tickets helps in resource allocation and planning decisions for achieving the desired objectives. The simulation results show how different work assignment policies play significant role in up-skilling and SLA success. We also demonstrate the importance of the novel service estimation model for practical use.

1 Introduction

A *Service System (SS)* is an organization composed of (a) the human resources who perform work, and (b) the processes that drive service interactions so that the outcomes meet customer expectations (Spohrer et al. 2007). Typically, a *service worker (SW)* represents a unit of human resource and a *service request (SR)* represents a unit of service work that (s)he is assigned. Hence, management of the SWs in service provider organizations is crucial. Over the past years, business and education groups have issued a series of reports indicating that due to rapid technological changes and increasing global competition, the skill demands of work are continually rising. Economists studying the changing workplace skill demands, have found that technological change is ”skill-biased” thereby increasing the demand for people who have multiple skills. Many businesses are asking employees to assume multiple roles and because of this shift, hiring has become difficult in countries in spite of steady unemployment rates.

This need for multi-faceted workers entails not only retaining the right skills, but also transforming the skills of the workers as dictated by the changing business requirements. For example, in the IT services domain, it may so happen that due to a transformation in the customer's environment, a provider has to quickly upskill his team. The current team of 10 people who only had expertise in the Solaris operating system needs to be transformed to a team where both the operating systems of Windows and Solaris need to be supported. While one option for the provider is to replace some of the Solaris personnel with new hires having Windows skills, a better option is to impart new skills to existing SWs such that they collectively meet the target skill requirements.

There are several approaches for imparting new skills: (a) class room training, where SWs dedicate training time for a certain duration and incur costs, (b) shadowing, where SWs observe the work of skilled SWs and learn, or (c) on-job training, where SWs pick up skills while actually doing the work. The nature of work in services involves substantial interactions not only with the customer but also with colleagues. Also, carrying out a task is far more difficult than simply knowing how to carry out a task. Hence, on-job training following minimal classroom training is the approach commonly adopted by service providers.

Very little understanding exists today on how the on-job training should be carried out. For example, how does the skill of a SW evolve when one or multiple new learnings are imparted ? Does this evolution of target skills change when (s)he already has some existing skills ? How do multiple learnings interfere with each other ? Can parallel learnings also reinforce ? Exactly how this on-job training should be planned and carried out such that impact to customer service in terms of service level agreement (SLA) is minimized ? All these form the pieces of what we call the on-job learning puzzle.

1.1 Problem Description:

The overarching objective of service organizations is to meet the service level agreements (SLA). They want to meet the quality goals without incurring too much additional costs. This makes it necessary to have a system in place which can adapt very quickly to changing client requirements and also being able to on board new clients quickly. Often, this translates to training mechanism for SWs which can make them come to speed quickly. This is because the proficiency of SWs in a skill impacts the service time of SRs requiring that skill which in turn has a direct impact on overall SLA success. When a SW goes through on-the-job training, then he is effectively getting to learn multiple skills over a period of time. This form of learning involves interleaved learning of skills because it is governed by incoming SRs. For example, the worker may end up working on 2 SRs that needs skill $s1$ followed by 4 SRs that need skill $s2$ and then 1 SR of skill $s1$. This means that there is not only the forgetting aspect that comes into play but also interference in learning. On-the-job training has always existed but usually is very ad-hoc and relies on simple heuristics when it comes to work assignment. In this work, we have modeled the training problem as a planning and scheduling problem. We have studied the evolution of SS under different scheduling policies to develop an insight into planning and scheduling strategies that can make on-the-job training more effective, efficient and sustainable.

The basic decision questions to be answered are i) whom should an SR be assigned once it arrives in the system with given set of SWs under given constraints , and ii) how to model service time estimations that imbibe realistic learning. In the case where all the software workers have the skills that are required to service the requests, the assignment problem can focus on the objectives of SLA. However, the situation is much more complex in the system where the workers have only basic level of skills to carry out the requests and the objective is that they will develop their skills as they service more and more requests. The learning is subjected to forgetting and interference which is a commonly observed phenomenon in real world. The cognitive theories of skill gain and loss are decades old (Newell and Rosenbloom 1981). Although descriptive and complete, these theories do not directly map to a computational understanding of skill gain and loss. On the other hand, in the industrial engineering field, mathematical models of learning and forgetting have been proposed and empirically validated (Jaber and Sikstrom 2004; Nembhard and Uzumeri 2000; Jaber, Kher, and Davis 2003). These models agree that the time taken to complete a unit of work is a function of the amount of current experience as well as a rate of learning. We draw on the LFCM model and adapt it to our setting (Jaber and Sikstrom 2004), thereby defining a novel model of service time estimation and skill evolution. Finally, we assume that SLAs are an accurate measure of service quality (Banerjee, Dasgupta, and Desai 2011). For example, as long as a provider completes 95% of all SRs received every month within 4 hours, the quality of service is deemed adequate. Our model entails a desired tension between the opposite forces of speed of achieving the target skills, and SLAs (as a measure of service quality). Since SRs assigned to SWs not possessing requisite skills would take longer to complete, the SLAs would be violated if too many work assignments involving skill gaps are made in a month. On the other hand, the opportunity to learn is also the greatest in the assignments involving skill gap, positively impacting the speed of achieving the target skills. The situation is complicated by the presence of interference between multiple skills while learning.

We have formalized the training problem as a Markov Decision Process and have proposed two heuristics to determine the single step transition probabilities. These heuristics, based on policies that serve as the guiding rules for work assignment, are employed for automated scheduling of SRs in such service systems. The quality of the two heuristics is compared by simulating them on various types of workloads and the evolution of the system is observed for a good duration of time. The real life handling of service requests is also modeled and compared with the two heuristics. The aim of service delivery industry is to dispatch the SRs in such a manner that it achieves required SLA success rate and up skilling of SWs at higher pace. The key goal requires SR allocation scheme to work in opposite directions. Increasing one may adversely affects other.

The results obtained clearly illustrate the irreconcilable tension between the goals of up-skilling and SLA. They go on to show that an insight into the interference levels between skills plays an important role in on-job training and how a disregard of that can deteriorate the performance of a service system that depends on such training mechanisms. The suitability of the two policies(heuristics) adopted while scheduling is also demonstrated for different situations.

2 Service Time Model and SLA

In an on-job training scenario when people need to learn multiple new skills, the gaps between the new and existing skills, the breaks between tasks requiring a particular skill as well as interference among the multiple skills, all affect the service time.

Learning Effect on Service Time: During on-job training, when lower skilled people do higher skilled work, service times get longer. Larger the difference between the skills, longer becomes the service times. This factor is modeled as gap learning factor or *glf*.

Forgetting Effect on Service Time: Manufacturing literature shows that (Jaber and Sikstrom 2004) time gaps or breaks between task executions causes forgetting. This has the effect of longer service times. Forgetting is proportional to the time gap (Jaber and Bonney 1996).

Interference Effect on Service Time: When a SW works on an assigned SR with particular skill requirements, working on these skills interfere with other skills exhibited by the SW. This interference results in lower recall accuracy of other skills.

We now present a service time model that takes into account the above three factors. This represents the skill progression model of a worker as multiple new learnings are imparted to her.

The time taken by a SW to complete an SR is stochastic and is shown (Banerjee, Dasgupta, and Desai 2011) to follow a lognormal distribution for a single skill. Let T_s be the service time required by a SW for a SR with particular skill requirement while working for n^{th} time on the same skill where the SW is working on the skill after $timeGap$ period. T_{BS} be base service time which denotes the time taken by service worker when working on the skill for the first time. Let $dist$ be the gap between required skill level of the SR and the current skill level possessed by SW. If the latter is higher or equal, $dist$ is zero. Otherwise it is given by the linear difference in the levels. Equations 1, 2 and 3 show the learning model while factoring in the time gap (Jaber and Bonney 1996) only. The learning factor (lf) is a constant (Jaber and Sikstrom 2004) which depends the learning pace of the SW.

$$\gamma = \frac{\log(1 + dist/timeGap)}{\log n} \tag{1}$$

$$glf = lf * (1 - \gamma) \tag{2}$$

$$T_s = T_{BS} * n^{1/glf} \tag{3}$$

Here, the learning factor is also a linear function of distance. The gap learning factor (glf) incorporates the lf and γ which is function of $timeGap$. There has been sufficient evidence in the literature to indicate that interference also causes forgetting. To include the interference in this model, we assumed that the effect of interference is equivalent to stretched time gap. To include the interference in this model, we used the results from (Das and Stuerzlinger 2013) that show that the effect of interference is equivalent to stretched time gap and modify the Equation 1 as Equation 4

$$\gamma = \frac{\log(1 + \frac{(dist + interferenceMeter)}{timeGap})}{\log n} \tag{4}$$

$InterferenceMeter$ keeps the track of number of times the SW has worked on other skills since last worked on the current skill. This meter is rest to zero every time when SW works on the skill.

SLA: SLA is specified by the customers for each incoming SR in terms of expected date of resolution. This is modeled in SS using timestamp according to the Equation 5.

$$SLA_{remain} = SR_{arrivalTime} + SR_{SLATime} - SR_{compTime} \tag{5}$$

where $SR_{arrivalTime}$ denotes the arrival timestamp of the SR, $SR_{SLATime}$ denotes the timestamp by which the SR should be completed in order to meet SLA, $SR_{compTime}$ denotes the time when the SR got completed and SLA_{remain} denotes the time remaining to meet the specified SLA.The positive value of SLA_{remain} indicates an SLA success otherwise SLA miss.The $SR_{compTime}$ is dependent on the expected service time of the SW who is working on it.

3 Work Assignment Strategy for On-Job Training

Given a set of initial resource profile that specifies resources' current skills and a target resource profile that specifies the number of resources required for each skill in the future workload, the transition in the resource profiles is brought about by means of work assignment that satisfies the desired goals of on-job training. Note that the new workload may have only partial overlap with the existing skills in the system. We assume, w.l.o.g., that the total number of resources required in the target profile is the same as existing numbers. The training problem is solved in two stages. The first stage is the planning problem that answers the question of which resource will be given on-job training on which skill. It is easy to see that many a times, resources will end up learning multiple new skills simultaneously. The second stage is the sequential scheduling problem of whom to assign an incoming request as and when it arrives.

3.1 Planning:

The planning problem is solved by adopting one of the following strategies i) Balance the new skills to be learned in terms of numbers ii) Balance the new skills to be learned in terms of interference. We explain both the strategies with a small example. Table 3.1 shows the current skills possessed by three SWs with id SW_1, SW_2 and SW_3 having 2 skills each. Last two columns in the table show the new skills required and the required number of SWs respectively. Here we can see that the skill id 5 is already present in the current skill profile. We also assume that the pair of skills $\{5, 7\}$ is highly interfering and other pairs are not interfering. If we follow the strategy (i), target skills are distributed in such a manner that each SW gets chance to learn equal number of new skill. Second column in Table 3.2 shows one such possible distribution where each SW requires to learn two new skills. However, according to strategy (ii), target skills are distributed by minimizing the interference among skills to be learned by a SW. Third column in the Table 3.2 shows one such possible distribution where no SW receives any interfering skill pair. We can observe that in strategy (i), the SW_3 has received skill 7 which is interfering with his existing skill 5, whereas in strategy (ii), SW_3 only gets one new skill to learn as he already knows skill 5 which is not interfering with existing skills.

Current Profiles		Target Requirements	
SW Id	Skill Ids	Skill Id	Requirement
SW_1	1,2	5	1
SW_2	2,3	6	3
SW_3	4,5	7	2

Table 3.1: SWs Current Profile and Target Requirements

	Balance New Skills	Balance Interfering Skills
SW Id	Skill Ids	Skill Ids
SW1	5,6	6,7
SW2	6, 7	6,7
SW3	6,7	5,6

Table 3.2: Skill Distribution Strategies

Such planning fixes the target skills for each resource and this information is used for the scheduling of SRs upon arrival. We assume that the planning problem is solved using one of the strategies and do not delve deep into the intricacies of the algorithm.

3.2 Scheduling:

The problem of on-job training is equivalent to multiple sequential invocations of the problem of scheduling incoming SR to appropriate SW such that the SLA target is met and up-skilling of all the service worker is maximized. This can be formalized as Markov decision process (MDP) by ensuring that decisions only depends on the current state. The problem of scheduling requires to design a policy that incrementally achieves the target skills in the shortest time. The policy prescribes the degree of suitability of an SR requiring a skill configuration to the many SWs possessing the many possible skill configurations.

MDP Formulation: We use MDP to model the sequential decision making. The objective of the decision process is to maximize an overall long-term performance in terms of SLA meet and up-skilling. Let (S, A, P, R) be a MDP, where S is a (in)finite set of states, A is a finite set of actions, P is probability that action a in state s will lead to state s' and R is the reward function. We formulate SR scheduling problem as follows:

State Space: The system state is represented as collection of states of each SW. Each SW's state consists of five tuples: shift availability, number of skills, skill level, current work load computed in terms of pending SRs in its queue, learning parameters as described in the equations 4, 2 and 3. Each time when a SW works on a SR, the learning parameters are updated and this results into new state for the SW. We define these collection of states as the state space for individual SW

$$S = \sum_{i=1}^{m}(S_i) \tag{6}$$

where S_i is the state space of i^{th} service worker. Since learning parameters get updated every time a SW resolves the assigned SR, this results in an infinite state space S_i because the parameters take real values which vary from one SW to another SW.

Action Space: The action space $A : \{a_i\}$ can be specified as finite set of actions, such as assign the SR to a particular SW, submitting the resolved SRs or resend the SR to global queue for re-work.

Reward Function: $R : S \times A = \mathcal{R}$ is reward function, which can be measured in our scenario in terms of improved learning model parameters such as reduced time gap, reduced interference and higher skill level and experience. In other terms we can say that improvement in the service time of the SW due to work on particular SR is reward for the SW. Whenever an action of resolved SR is taken, the SW get rewards based on the time taken to resolve the SR and his/her current average service time.

Transition Probability: The probability of a SW of being assigned incoming SR is transition probability. Transition probability is an outcome of the scheduling policy described below. In order to maximize the SLA success rate and up-skilling rate, we define two different policies which concentrate on one goal at a time. A policy results in a set of eligible SWs for a SR and one of them is chosen at random. The transition probabilities to the remaining SWs is made 0. In the evaluation section, we compare these policies with the naive policy which assigns SR based on the first available scheme.

SLA First Policy: SLA First scheme aims to maximize the SLA success, based on the observation that service worker w with least load, denoted by $minLoad$, is able to quickly complete the assigned SR, hence maximizing the SLA success. Each SR in the SW's queue contributes to load proportional to the skill level gap between incoming SR and the SW. A threshold on the queue load determines if a SW is overloaded. Algorithm 1 formally describes the SLA First scheme for assigning a service request SR to assign appropriate service worker w among the pool of available service workers $SWList$. Initially SLA First schemes checks for all the service workers which are having same skill level as required by the SR, are not overloaded and can meet SLA based on expected service time. Among them, it finds the least loaded service worker w. The load due to pending SRs in the queue is denoted by $SRPendingQueueLoad$. If all the service workers with equal skill level as required by SR are overloaded or not available, then the scheme looks for the service workers having the same skill required by the SR with one level lower and higher which are not overloaded and so on. As we have finite skill levels, the algorithm terminates. If it does not find anyone, then the least loaded SW is chosen. Amongst the shortlisted SWs, it then computes γ to find who has the maximum learning potential.

```
Input: SR, SWList
Output: SW_id
id = φ
minLoad = 150
diff = 0
while id = φ AND diff < 4 do
    for each w_i ϵ SWList do
        if abs(SR_SkillLevel − w_i.SkillLevel) = diff
        AND w_i.overload = false then
            if
            minLoad > w_i.SRPendingQueueLoad
            then
                id = w_i.id
                minLoad =
                w_i.SRPendingQueueLoad
            end
        end
    end
    diff = diff + 1
end
return id
```

Algorithm 1: SLA FIRST POLICY OUTLINE

Learning First Policy: Learning First scheme gives more chances to the service workers with lower skill levels in order to assign them more service requests and increase their experience and learning. This scheme looks at all the service workers which can complete the SR and meet SLA by calculating the expected service time and check if it is less than

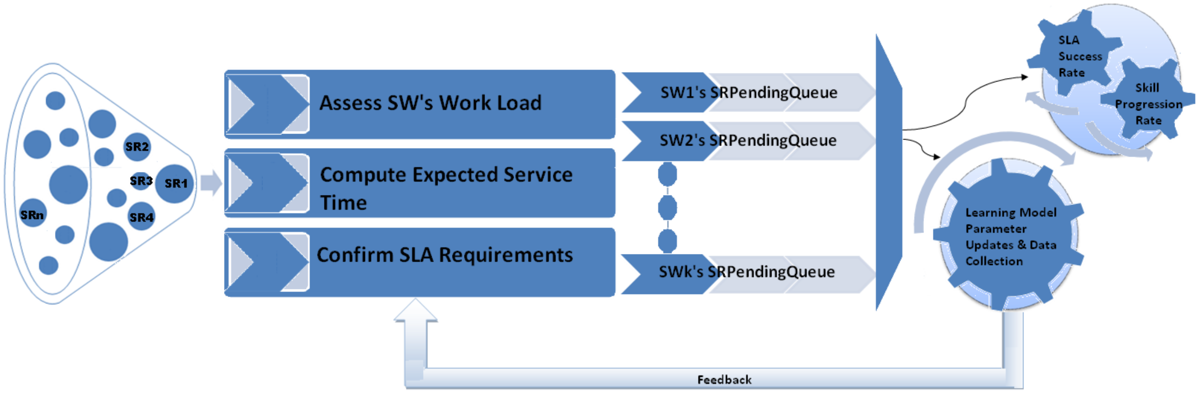

Figure 1: Discrete Event Simulation of Service System On-Job Training

the remaining SLA time. Since this scheme always prefers the service worker with maximum worst case service time $maxWorstServiceTime$ who can complete the SR within SLA, there are higher chances of increasing the skill level of the service worker at the cost of increasing the probability of missing SLA. Algorithm 2 formally describes the Learning First scheme for assigning a service request SR to assign appropriate service worker w among the pool of available service workers $SWList$. Initially Learning First schemes checks for all the service workers which have expected service time less than remaining service time and not overloaded. Among them, it finds the least loaded service worker w with highest value of worst case service time $maxWorstServiceTime$.

Input: $SR, SWList$
Output: SW_{id}
$id = \phi$
$minLoad = 150$
$maxWorstServiceTime = 0$
for *each* $w_i \epsilon \ SWList$ **do**
  if $w_i.expectedServiceTime < SR.remainingSLA$ $AND \ w_i.overload = false$ **then**
    if $minLoad > w_i.SRPendingQueueLoad$ $AND \ maxWorstServiceTime <$ $w_i.worstServiceTime$ **then**
      $id = w_i.id$
      $minLoad = w_i.SRPendingQueueLoad$
      $maxWorstServiceTime =$ $w_i.worstServiceTime$
    end
  end
end
return id

Algorithm 2: LEARNING FIRST POLICY OUTLINE

3.3 Policy evaluation:

The SR arrivals from customers, their assignments to SWs according to the policy, skill gains and losses, and SLA measurements are simulated via a discrete event simulation model (Banerjee, Dasgupta, and Desai 2011). A policy remains in force for the period of simulation (say, a month). Such simulations can be run for a real life SS to learn which is the best policy for the forecasted demand. If the demand requires learning lot of new skill that are interfering, then the policy of work assignment would be different from the kind of demand where no new skills have to be learnt and the only objective then becomes that of meeting SLAs. The details of the simulation framework and the evaluation of the different policies under different types of workload is presented in the following sections.

4 Simulation Framework

Simulation framework models the Service System. A service request(SR) arrives in the system and is redirected to a service wroker (SW) who resolves it. A lot of decision making happens between these two steps and state of the SR and SW changes dynamically. A valid set of states for a SR is {Arrived, Queued, Pending, InProcess, Serviced, Failed, Resubmitted} and similarly, set of valid state of service worker is defined as {Available, NotAvailable}. Expected service time of a SR for each SW computed using the learning curve described in section 2.

4.1 Workload Generation and Distribution parameters:

Arrival rate and parameters: Given the average inter-arrival time, assuming Poisson distribution, the workload is generated for a specified number of weeks. For each arrival, there are associated parameters of priority, skill and skill-level required to resolve the SR. Here we assume that a SR is assigned to only one SW and that SWs can have multiple SRs pending in its queue with at most 5 SRs, with skill gap 0, at any given instance of time during simulation.

Skill Requirement: For a particular skill, the number of required SWs is specified as input. This is a variable parameter, which we will change in order to analyze the effect of number of skills to be learned by a SW on SLA success and skill progression rate. A SW can be at one of the following level for a skill { Expert, Advanced, Developed, Basic }. We have adopted a simplified model of skill levels as proposed in (Dreyfus and Dreyfus 1980). In simulation, as time

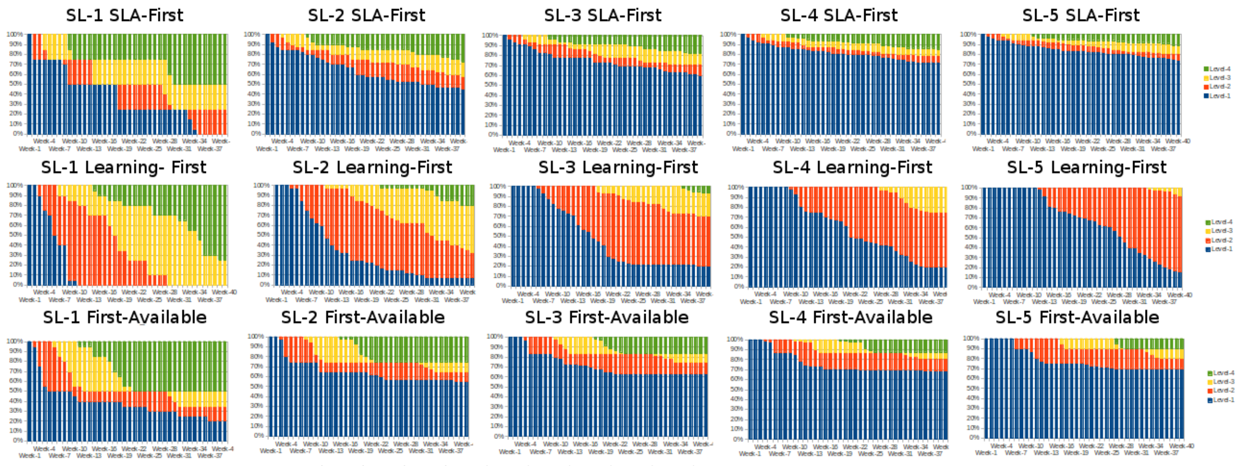

Figure 2: Effect of Skill Load on skill progression

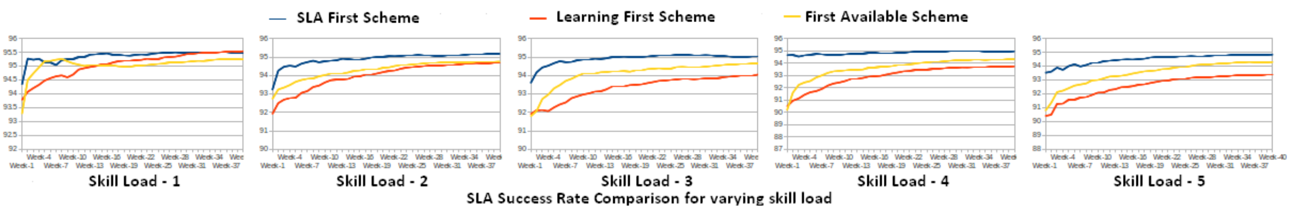

Figure 3: Effect of Skill Load on SLA

elapses and SW gains experience with skills, SW's skill level is increased from Basic to Developed, and subsequently to Advanced and Expert.

Skill load per SW: Based on the skill set requirement and available resources, skill load per SW is assigned with varying distribution such as uniform, left skewed and right skewed skill load distribution. Under uniform skill load distribution, each SW gets equal number of skills where as in left (right) skewed skill load distribution, most of the SWs get less (more) number of skills. With various distributions, skill load per SW is given as input to the simulation framework and relative effect on the SLA success and skill progression is measured.

Interference load: Two skills are considered either interfering or non-interfering. In case of interfering, every interfering pair is assumed to interfere in the same manner. In real life, it is possible that one pair is more interfering than the other pair. However, there is no quantitative model for this and hence, we assume that whenever interference exists, it affects in the same quantum. Interference load is analogously defined as skill load. Based on the combination of skill load and interference among skills, we arrive at four classes of workload as follows:

1. Left Skewed Interference Load: Majority of SWs have high interference load, while skill load is normally distributed.
2. Right Skewed Interference Load: Majority of SWs have low interference load, while skill load is normally distributed.
3. Left Skewed Skill Load: Majority of SWs have high skill load, while interference load is normally distributed.
4. Right Skewed Skill Load: Majority of SWs have low skill load, while interference load is normally distributed

4.2 Modules:

The main modules of simulation framework as in Figure **??** are described below.

Global queue of incoming SRs: A global queue is maintained which accepts all the incoming SRs with different priority and different skill requirement. For every SR, we maintain the information such as priority, SLA deadline, skill and corresponding level requirement, status as $inqueue$ (default), $pending$, $inservice$, $rework$ or $completed$. This global queue serves as the input to next phase which is dispatching module.

Dispatching Module: Identify appropriate SW using policies: The module accepts SRs from global queue one by one and uses list of all the SWs in order to search for the most suitable SW for the current SR for assignment. For each SW, we maintain information such as existing skills, new skills being learnt, shift availability, overload status. This phase uses one of the scheduling policies as described in section 3. After identifying the SW, SR is sent to its queue and the status of the SR changed from $inqueue$ to $pending$. Also the $SRPendingQueueLoad$ is also updated as described in equation 7.

SR's Skill Level and Queue load : A service worker's queue can have SRs of skill levels different than his current level. The load value in such a situation is normalized by having more complex SRs contribute more to the load than the lower level ones. We assume the normal load of a SR for SW is equal to 20 and the Equation 7 is used to calculate the load due to higher level ones.

$$weight = 20 + (SR_{skill-level} - SW_{skill-level}) \times 5 \quad (7)$$

Let $curload$ of a SW denote the load due to pending SRs in the queue. We calculated whether a SW is overloaded or not as follows:

$$overloaded = \begin{cases} yes & \text{if } curload \geq 100 \\ no & \text{otherwise} \end{cases}$$

Learning parameter updation and statistical data collection: Once SR is assigned to a SW, it remains in the queue of the SW till all the SRs which arrived before it are not attempted to resolve by the SW and status is updated to $pending$. When SW works on the SR, the status is updated to $inservice$. The experience gained by working on SR of a particular skill is captured through learning parameters as explained below.

1) For simulation purpose, we set the learning factor to 0.1. Learning decreases by 0.01 as distance increases by 1 in the learning formula.

2) Additive interference and interferenceMeter: We assume the additive model of interference where each interfering skill adds a unit of interference into interferenceMeter independent of other interfering skills. We start an interferenceMeter for each skill for every SW with value 0. Whenever a SW works on a particular skill, it increases the value of interferenceMeter of skills being interfered by current skill. InterferenceMeter value for current skill is used to update SWs learning curve parameters and reset to 0.

The framework also continuously collects data such as skill level progression rate, SLA success rate. Skill level $level$ is upgraded after sufficient experience for the skill. In our simulation model, we assume that after working on 500 SRs of the same skill, SWs skill level is incremented by 1 with minimum value 1 and maximum value 4. These skill level from 1 to 4 represents the proficiency of SW as Basic, Developed, Advanced and Expert respectively. After servicing the SR, it is sent to next phase which performs check if rework is required and collect the data and update learning model parameters.

5 Experiments and Evaluation

We have additionally modeled a *first available policy* that works on the sole objective of assigning work to the SW who is least loaded. This has been done to model the policy that seems to be widely prevalent in the SS for work assignment. This will help to compare the strengths and weaknesses of the proposed heuristics. Note that first available policy does not compute expected service time while choosing the SW as is done by SLA first policy thus differing from it. We study the two novel heuristics of work scheduling under different types of workload. The input to the planning stage is the current skill profile and target skill profile based on the workload. The output of the planning stage can result in one of the following types of learning:

I) Balanced skill learning load : This implies that most resources have to learn the same number of skills. We have considered two sub cases within this: i) The skills are interfering with each other, ii) The skills are non interfering.

II) Skewed skill learning: The learning can be skewed with respect to number of skills to be learned or with respect to skill interference. This basically translated to the following cases: i) The number of skills to be learned is left skewed or right skewed assuming constant interference ii) The interference is either left skewed or right skewed assuming balanced skill load.

5.1 Effect of skill learning load on SLA rate and Skill progression:

The first set of simulations were performed on a target resource profile with balanced skill learning load. In figure 2, each column corresponds to different skill load starting from left most column with skill load 1 to right most column with skill load 5. Each row represents different policy implemented. First row shows the results for SLA first policy, second row for learning first policy and third row for first available policy. Color transition of blue to green in the graph shows number of different skill levels SWs in a specific week. x-axis shows the time line of 40 weeks. Following are the key observations made for both types target resource profiles ,that is, balanced learning load with interfering and non-interfering skills:

1. The increase in the number of skills to be learned slows down the learning rate considerably. This is irrespective of whether the skills are interfering or not. The charts in figure 2 are for interfering skills. We observed that if the skills are non-interfering then, also the patterns remain same with gradient of learning becoming steeper.
2. Learning first policy performs the most uniform upskilling. It gives more chances to lower skill workers wherever possible at the risk of SLA. It up-skills most uniformly and tries to up-skill all the SWs at equal level. SLA first policy and first available show a similar trend of non-uniform learning which is very pronounced in the chart with skill load > 3. We observe that these two policies support quick learners by assigning them more and more work. This happens because when some SWs reach higher level, they quickly resolve the SRs and become available early as compare to lower level SWs, hence get more SRs and progress quickly while others remain at lower level because of lack of chance to get work on incoming SRs.

In figure 3, SLA success rate for simulation runs on same input data as that of figure 2 is shown. It is quite clear that SLA first policy is best in meeting SLA followed by first available and learning first policies. The SLA first policy outperforms first available because it considers the expected service time based on the model discussed in section 2 for each incoming SR before choosing the SW. In doing so, it ensures that more suitable SW is chosen based on SLA vs. expected service time computation after inspecting the queue load.

5.2 Effect of skewed learning on SLA rate and Skill Progression:

In figure 4, first chart shows the results when the majority of the SWs have high skill load which are not interfering

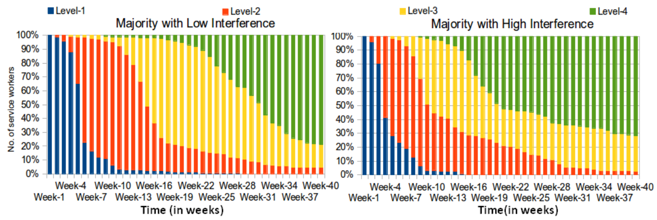

Figure 4: Skewed Skill Load and Varying Interference Load

whereas second chart contains the results when majority of the SWs have high skill load which are interfering. We observed that initially there was higher progression rate in scenario 2 where most of the workers are having high interfering skill load. But when most of the SWs progressed to higher level, their service time improved, they resolve SRs quickly and become eligible to resolve new SRs. Hence, they get more and more SRs, which in turn increases the interference and slows down the learning. In scenario 1, where majority of SWs are having high non-interfering skill load, initial skill progression rate is slower and increases exponentially towards the end and overall it performs better than scenario 2 in terms of skill progression. The skill load distribution for these two scenarios are as follows: 1. Column-1: 10% workers with 2 skills, interfering with each other and 90% worker with high skill load of 4 skills, non interfering 2. Column-2: 20% workers with 2-3 skills, interfering with each other and 80% worker with high skill load of 4 skills, non interfering

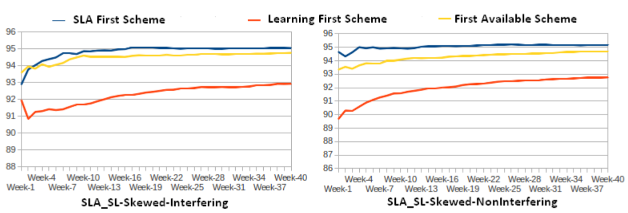

Figure 5: Effect of Skewed learning load on SLA

The simulations on skewed load show the impact of interference on learning and that it can alter the learning rates irrespective of the policy. However, the learning first policy continued to perform better than the others for such workloads as well in terms of uniform learning. However SLAs get adversely affected as shown in figure 5.

6 Related Work

In this section, we situate our work within prior research on team and organizational learning theories, resource planning, human skill evolution and learning. We also talk about work in related domain other than the ones already discussed in Section 1.

There has been a significant body of work focused on teams and their learnings. About two decades back researchers(Williamson 1981; F. M. Jablin and (Eds.) 1986) studied the effects of organizational structure (i.e. hierarchy, team etc.) on metrics like problem solving, cost, competition and drive for innovation and also the effect (Carley 1992) of learning and turnover on different structures. At the same time, collaborations and communication with teams have also seen a comprehensive body of research. Carley's (Carley 1991) theory of group stability postulates a relationship between individual's current knowledge and her behavior. She also found that a group's interaction increases as commonality across knowledge dimensions increases. Very recently (Liemhetcharat and Veloso 2012) presented the notion of synergy in human teams or *how well* they work together. The authors defined an algorithm to compute the optimal team for a task, using the synergy graph.

Learning has also been looked at in the context of human resource planning (Bordoloi and Matsuo 2001), (Bordoloi 2006), where there is a need to forecast the future skill mix and levels required, as well as in context of dynamic environments like call centers(Gans and Zhou 2002), where both and learning and turnover are captured to solve the long and medium term staffing problem.

In context of skill evolution, Dibbern et. al (Dibbern and Krancher 2012) captures the dependencies of expertise, task complexity, support information and learning tasks on learning effectiveness during KT. Imparting knowledge with on-the-job training has also been another popular method for imparting skills. Work in labor economic theory (Barron, Black, and Loewenstein 1989) has attempted to assess how much on-the-job training is needed for a specific worker, based on his current expertise and learning ability.

In the domain of learning, authors (Knox and Stone 2012) talks about accelerating learning of agents via human feedback. It is also shown that (Urieli et al. 2011) optimizing skills in isolation does not necessarily benefit their combined operation. Guadagnoli et. al (Guadagnoli and Lee 2004) formulate a challenge point framework for motor learning where the learning is maximized at an optimal challenge point. According to authors, how much an individual learns when challenged, depends on the skill level of the performer and the task complexity. Apart from the learning and forgetting models ((Jaber and Sikstrom 2004; Jaber, Kher, and Davis 2003; Nembhard and Uzumeri 2000; Sikstrom, , and Jaber 2002)) presented in Section 1, recent work (Subagdja et al. 2012) presents interesting results on how memory consolidation and forgetting processes regulate the memory capacity, and can mutually improve the effectiveness of learning.

7 Conclusion

The simulations of work scheduling heuristics with different settings show that there is trade-off between SLA success and up-skilling. The presence of interference slows down the learning rate. So does the number of skills to be learnt. We saw that for same number of skills to be learnt, the learning is slow if the skills are interfering. Thus, the advisable policy for the planning stage should be such that the new skills to be learnt are spread out evenly and in a manner that they are not interfering. If interference cannot be avoided, then an attempt should be made to minimize the interfering skills to be learnt per resource. As far as the scheduling is concerned, learning first policy should be adopted if SLAs are relaxed and uniform learning is more desirable. To promote a competitive environment, SLA first policy is most advisable.

References

Banerjee, D.; Dasgupta, G.; and Desai, N. 2011. Simulation-based evaluation of dispatching policies in service systems. In *Winter simulation conference*.

Barron, J.; Black, D. A.; and Loewenstein, M. A. 1989. Job matching and on-the-job training. *Journal of Labor Economics* 7(1):1–19.

Bordoloi, S. K., and Matsuo, H. 2001. Human resource planning in knowledge-intensive operations: A model for learning with stochastic turnover. *European Journal of Operational Research* 130(1):169–189.

Bordoloi, S. 2006. A control rule for recruitment planning in engineering consultancy. *Journal of Productivity Analysis* 26(2):147–163.

Carley, K. 1991. A Theory of Group Stability. *American Sociological Review* 56:331–354.

Carley, K. M. 1992. Organizational learning and personnel turnover. *Organization Science* 3:20–46.

Das, A., and Stuerzlinger, W. 2013. Unified modeling of proactive interference and memorization effort: A new mathematical perspective within act-r theory. In *Proceedings of the annual meeting of the cognitive science society - 2013*, COGSCI '13, 358–363.

Dibbern, J., and Krancher, O. 2012. Individual knowledge transfer in the transition phase of outsourced softwaremaintenance projects. In *ISB-IBM Service Science Workshop*.

Dreyfus, S. E., and Dreyfus, H. L. 1980. A Five-Stage Model of the Mental Activities Involved in Directed Skill Acquisition. Technical report.

F. M. Jablin, L. L. Putnam, K. H. R., and (Eds.), L. W. P. 1986. *Handbook of Organizational Communication: An Interdisciplinary Perspective*. Sage.

Gans, N., and Zhou, Y.-P. 2002. Managing learning and turnover in employee staffing. *Oper. Res.* 50(6).

Guadagnoli, M. A., and Lee, T. D. 2004. Challenge point: a framework for conceptualizing the effects of various practice conditions in motor learning. *Journal of motor behavior* 36(2):212–224.

Jaber, M. Y., and Bonney, M. 1996. Production breaks and the learning curve: The forgetting phenomenon. *Applied Mathematical Modelling* 20(2):162 – 169.

Jaber, M. Y., and Sikstrom, S. 2004. A numerical comparison of three potential learning and forgetting models. *International Journal of Production Economics* 92(3).

Jaber, M. Y.; Kher, H. V.; and Davis, D. J. 2003. Countering forgetting through training and deployment. *International Journal of Production Economics* 85:33–46.

Knox, W. B., and Stone, P. 2012. Reinforcement learning from simultaneous human and mdp reward. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '12.

Liemhetcharat, S., and Veloso, M. 2012. Modeling and learning synergy for team formation with heterogeneous agents. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '12, 365–374.

Nembhard, D. A., and Uzumeri, M. V. 2000. Experiential learning and forgetting for manual and cognitive tasks. *International Journal of Industrial Ergonomics* 25:315–326.

Newell, A., and Rosenbloom, P. 1981. Mechanisms of skill acquisition and the law of practice. *Cognitive skills and their acquisition* 1–55.

Sikstrom, S.; ; and Jaber, M. Y. 2002. The power integration diffusion (pid) model for production breaks. *Journal of Experimental Psychology* 8:118–126.

Spohrer, J.; Maglio, P.; Bailey, J.; and Gruhl, D. 2007. Steps toward a science of service systems. *Computer* 40(1):71–77.

Subagdja, B.; Wang, W.; Tan, A.-H.; Tan, Y.-S.; and Teow, L.-N. 2012. Memory formation, consolidation, and forgetting in learning agents. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '12.

Urieli, D.; MacAlpine, P.; Kalyanakrishnan, S.; Bentor, Y.; and Stone, P. 2011. On optimizing interdependent skills: A case study in simulated 3d humanoid robot soccer. In *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*.

Williamson, O. E. 1981. The economics of organization: The transaction cost approach. *American Journal of Sociology*.