Service Systems with Postponable Acceptance: Multi-project Selection and Assignment Problem

Keumseok Kang, George Shanthikumar, and Kemal Altinkemer

Krannert School of Management, Purdue University, West Lafayette, IN 47906

July 2011

ABSTRACT

We study the dynamic order acceptance and resource assignment problem of a service system which provides heterogeneous services using heterogeneous servers. Unlike traditional acceptance and/or assignment problems, decision makers in our settings can strategically postpone acceptance decisions anticipating more profitable orders to arrive, with the risk of lost orders and low resource utilization. We formulate this problem using a dynamic and stochastic programming approach, find the structural properties of the optimal policy, propose approximate policies, and conduct extensive computational experiments to show the effectiveness of proposed policies. This problem is motivated by the multi-project selection and assignment problem of a software organization; however, it can be applied to general service systems where postponable acceptance decisions are allowed.

Keywords: service systems, heterogeneous services and servers, postponable acceptance, optimal policy, dynamic and stochastic programming, approximate policy.

Please do not quote and distribute without the permission of authors
1 Introduction

In this paper, we study the dynamic order acceptance and resource assignment problem of a service system which provides heterogeneous services with heterogeneous servers. One of unique characteristics of this problem is that the decision maker can accept orders strategically by postponing order acceptance decisions to some extents while waiting for more profitable orders. In other words, unlike traditional acceptance decision problems such as the secretary problem, online-Knapsack problem, and general queuing problem, in our problem, the acceptance (or reject) decision for an order does not have to be made upon the arrival (i.e., request) of that order. Instead, the decision maker may defer the acceptance decision as long as that order is available to accept (i.e., until that order is withdrawn by the customer). By doing so, the decision maker may improve its profits by strategically waiting for and accepting more profitable orders. However, at the same time, the decision maker also faces the risk of losing current orders in hands and low resource utilization due to the uncertainty about new order arrivals and current order abandonment. For example, more profitable orders may not arrive until the postponed order is cancelled by customer. This situation may also result in unnecessary idle resources and finally may harm profits.

This study is motivated by the software development project selection and resource assignment problem of IT services companies. Let us explain the problem more. Basically, in the IT services industry, customers order software development services to an IT service company, and then the IT services company accepts or rejects the customers’ orders based on the capability and availability of the company’s resources and the profitability of the orders. After that, the company provides software development services for the accepted orders. From the perspective of an IT services company, it is generally true that the more requests to accept, the more profits to make. However, due to the limited resources and the variety of services being requested by customers, the company cannot accept all orders requested by customers. Instead, the company has to selectively accept some orders and reject some others. When accepting or rejecting an order, since contracting a software development service is a critical business decision to the IT service company, a period of time is usually given to the IT services company for making a decision – accept or reject the order. In other words, the company does not have to decide whether or not to accept a customer’s order upon its arrival, but may defer the decision to some extent (e.g., for several days or weeks). However, the company also should make a decision in a timely manner because customers usually contact more than one IT services company, and one of competitors may take this opportunity before the company takes it. It means that the company basically does not know until when the request is available to accept (i.e., uncertainty of order abandonment). In addition, IT
services companies generally suffer from the uncertainties about new customer orders – when, what, and how many new orders arrive – and the uncertainties about internal resources – when the current jobs are finished so that the resources can be available to assign new jobs. The objective of an IT services company in this problem is to maximize its profits by accepting right orders and assign them to right resources over a given decision horizon (e.g., one year).

Although we are motivated from a particular problem in software services, similar problems can be found in other domains where different types of orders are come from outside and deferred acceptance/reject decisions for those orders are allowed. Examples may include accepting patients for surgeries in hospitals, recruiting students in colleges, accepting make-to-order requests in manufacturing companies, etc. Moreover, we believe that postponable decisions are not only applicable to the acceptance problems but also can be applied to any decision-making problems where decisions are associated with perishable or limited resources and tradeoffs exist between the benefits obtained by waiting for preferable future alternatives and the risk of losing the current options. Such problems can be generally found in the areas of revenue management, financial engineering, and resources management.

The remainder of the paper is organized as follows. In the next section, we review the related literature. Then, we explain our problem definition and formulation. This is followed by the analysis for the optimal policy. Then we propose simple approximate policies and report the computational experiment results. Finally, we conclude with a discussion of the results, contributions, and limitations of this study.

2 Literature Review

One of the most similar problems in the literature is the call routing problems in call centers. In a call center, customers’ calls are stored in queues and served (usually in a “first comes first served” manner) when available agents are revealed in general. Like ours, some prior studies in this area also assume that customers are impatient and so some calls in queues may be abandoned before they are served (Atar et al. 2004; Bassamboo, Harrison, and Zeevi 2005; Harrison and Zeevi 2004; Jouini et al. 2010; Koole et al. 2003; Tezcan and Dai 2010) and some studies also assume heterogeneous services and servers (Bassamboo, Harrison, and Zeevi 2005; Harrison and Zeevi 2004; Tezcan and Dai 2010). However, our problem is different from these call center studies in a sense that acceptance decisions and assignment decisions are separated in our model, while both decisions occur at the same time in the call center routing problem. In other words, in the call center routing problem, there is only one type of queue – waiting for assignment. Customer calls in those queues are accepted and assigned when agents become available and start being served. On the other hand, in our problem, there are two types of queue: waiting
for acceptance and waiting for assignment. Like call centers, incoming orders are first stocked in queues waiting for acceptance. However, unlike call center models, orders are stored in queues waiting for assignment after being accepted. For this reason, unlike the call center routing problem, in our model, acceptance decisions are somewhat more strategic and separated from assignment decisions. This means that in order to preempt more profitable orders or hedge the risk of low resource utilization, acceptance decisions can be made even before any available resources are revealed or even before alternative orders arrive.

Another similar problem is the stochastic (or online) Knapsack problems. Most stochastic (or online) Knapsack problems assume the total number of items which will arrive is given a priori. However, similar to ours, some studies assume that there is a decision horizon (i.e., time deadline) and items arrive according to a random process (e.g., Poisson process) during the decision horizon (Kleywegt and Papastavrou 1998; Papastavrou et al. 1996; Ross and Tsang 1989; van Slyke and Young, 2000). Although

Figure 1. Call center routing problem vs. Our problem

Another similar problem is the stochastic (or online) Knapsack problems. Most stochastic (or online) Knapsack problems assume the total number of items which will arrive is given a priori. However, similar to ours, some studies assume that there is a decision horizon (i.e., time deadline) and items arrive according to a random process (e.g., Poisson process) during the decision horizon (Kleywegt and Papastavrou 1998; Papastavrou et al. 1996; Ross and Tsang 1989; van Slyke and Young, 2000). Although
most stochastic Knapsack problems generally assume that resources are assigned to jobs only once during the decision horizons, Ross and Tsang (1989) assume that resources can be reassigned many times as long as those resources complete assigned jobs. Ross and Tsang (1989)’s model is similar to our model; however, it is different from ours in that it does not allow postponable acceptance decisions.

Our problem is also similar to the secretary problem (Freeman 1983 and Ferguson 1989). The secretary problem is a special case of online or stochastic Knapsack problem of which resource requirements are all same. Like the Knapsack problem, a general assumption for the timing of acceptance decisions in the secretary problem is being immediate upon the arrival of an order, which is mainly different from ours. Also, like the Knapsack problem, there are no multiple assignments per resource in the secretary problem. If all secretaries are selected once, they are not replaced or reselected. See figures 1 for the comparisons between the call center routing, online Knapsack, and secretary problem as well as and our problem.

Lastly, similar to our study, there are prior attempts to find the structural properties of the exact policy for the acceptance and assignment problems of services system with heterogeneous resources (e.g., Ahn et al. 2005; Akcay et al. 2010; Xu et al. 1992). However, no single study is identical to ours. For example, Ahn et al. (2005) and Xu et al. (1992) did not incorporate the concept of postponable decisions in their model. Akcay et al. (2010) studied the most similar problem, but their problem does not have the process of service completions and reassignments. In other words, their assignment is one-time decision while ours is dynamic assignment decisions.

In summary, to the best of our knowledge, our model is unique and has not been addressed yet in the literature.

3 Problem Definition and Formulation

In this section, we define the problem and formulate it as a dynamic and stochastic optimization model. Consider a service system where service orders arrive at discrete times \{1, 2, \ldots, n\}. Let period \(k\) be defined as the time interval \([k, k + 1)\). There are \(I\) service types; and, each order belongs to one service type \((i = 1, 2, \ldots, I)\). There are \(J\) server types \((j = 1, 2, \ldots, J)\); and, each type has the maximal capacity, \(b^j > 0, j = 1, 2, \ldots, J\). All servers are cross-trained so that they can provide any type of services but at different costs, \(c_{kj}^i\).

Let \(Z_{k}^i\) be the total amount of \(i\)-type orders which arrive at the end of period \(k - 1\) (i.e., at time \(k\)), \(k = 1, 2, \ldots, n\). Let \(X_{k:2}^i\) be the total amount of \(i\)-type orders, including the ones which just arrived at the end of period \(k - 1\), for which the acceptance has been postponed by the end of period \(k - 1\). Let \(X_{k:1}^i\) be
the amount of orders which are in the queue for assignment but has not been assigned to any server yet by the end of period $k - 1$.

At the beginning of period $k$, some part of orders waiting in the queue for acceptance, $X_{k:2}^i$, can be accepted or rejected, or any part of them can be postponed. Let $A_k^i$ and $R_k^i$ be the amount of orders accepted and rejected, respectively at the beginning of period $k$. Similarly, some part of order in the queue waiting for assignment can be assigned to servers. Let $U_k^{ij}$ be the amount of $i$-type orders which are assigned to $j$-type servers at the beginning of period $k$. Acceptance and assignment decisions are made almost at the same time; therefore, some orders can be assigned to servers immediately after they are accepted, without staying in the queue for service commencement. Thus we have

$$X_{k+1:1}^i = X_{k:1}^i + A_k^i - \sum_{j=1}^{J} U_k^{ij}, \quad k = 1, 2, ..., n, \ i = 1, 2, ..., I$$
At the end of period $k$, following events may occur:

E1) Some services (a fraction, $L_k^i \in [0,1]$) are completed: $Q_k^i L_k^i$

E2) Some unaccepted orders (a fraction, $W_k^i \in [0,1]$) are abandoned: $Y_k^i W_k^i$

E3) New service orders arrive: $Z_{k+1}^i, Z_{k+1}^2, ..., Z_{k+1}^j$

At the beginning of period $k$, following decisions are made:

D1) Accept and reject orders: $A_k^i, R_k^i$

D2) Assign orders: $U_k^i$

$(X_k^i, X_{k+1}^i, G_k^i) \rightarrow (Y_k^i, Y_{k+1}^i, Q_k^i)$

$Y_k^i = X_k^i + A_k^i - \sum_{j=1}^{j} U_{k}^j$

$Y_{k+1}^i = X_{k+1}^i - A_k^i - R_k^i$

$Q_k^i = G_k^i + U_k^i$

At time $k+1$,

$X_{k+1}^i = Y_k^i$

$X_{k+2}^i = Y_{k+1}^i (1 - W_k^i) + Z_{k+1}^i$

$G_{k+1}^i = Q_k^i (1 - L_k^i)$

Figure 3. Dynamics of the problem

The capacity of $j$-type servers which are currently serving $i$-type orders at the beginning of period $k$ is $G_k^{ij}$. Therefore, the capacity of available (i.e., idle) $j$-type servers, $S_k^j$, is $b^j - \sum_{i=1}^{i} G_k^{ij}$. The total amount of orders newly assigned to $j$-type servers should not be larger than the capacity of available servers (i.e., $\sum_{i=1}^{i} U_k^{ij} \leq S_k^j, j = 1, 2, ..., J$). Sometimes, some servers may not be unassigned and remain idle. This unassigned capacity is rolled over to next period. A fraction ($L_k^{ij}, 0 \leq L_k^{ij} \leq 1$) of non-idle (i.e., working) servers complete their assigned orders at the end of period $k$ and become available for new assignments at the beginning of period $k + 1$. Thus we have

$G_{k+1}^{ij} = (G_k^{ij} + U_k^{ij})L_k^{ij}$

$S_{k+1}^j = b^j - \sum_{i=1}^{i} G_{k+1}^{ij}$

Customers who place orders may be impatient, and some part of orders waiting in the queue for acceptance may be abandoned. At the end of period $k$, only a fraction ($0 \leq W_k^j \leq 1$) of orders which have not been accepted at the beginning of period $k$, remain waiting for acceptance, while the others are abandoned. Thus we have
Accepting orders are associated with increasing revenue. Let $r_k^i$ be the revenue generated when an $i$-type order is accepted at time $k$. Waiting and providing services incur associated costs. Let $c_k^{ij}$ be the service cost used for a $j$-type server to serve an $i$-type service during period $k$. Let $b_{k:1}^i$ and $b_{k:2}^i$ be the penalties for letting an order to wait during period $k$, in the queues for service commencement and acceptance, respectively. We assume that all costs accrue at the beginning of each period.

Let $X_{k:1} = (X_{k:1}^1, X_{k:1}^2, ..., X_{k:1}^l)$, $X_{k:2} = (X_{k:2}^1, X_{k:2}^2, ..., X_{k:2}^l)$ and $G_k = (G_k^{11}, G_k^{12}, ..., G_k^{ll})$. Then the state variables at time $k$ is denoted by $(X_{k:1}, X_{k:2}, G_k)$, and the expected total reward received during the $n$ periods is

$$
\mathbb{E}\left[ \sum_{k=1}^{n} \sum_{i=1}^{l} \left( r_k^i A_k^i - b_{k:1}^i \left( X_{k:1}^i + A_k^i - \sum_{j=1}^{l} u_k^{ij} \right) - b_{k:2}^i (X_{k:2}^i - A_k^i - R_k^i)W_k^i \right. \\
\left. - \sum_{j=1}^{l} c_k^{ij} (q_k^{ij} + U_k^{ij}) \right) + \mathbb{E}[v_{n+1}(X_{n+1:1}, X_{n+1:2}, G_{n+1})] \right]
$$

Let $x_k = (x_{k:1}, x_{k:2}, g_k)$ represent a state before making decisions – accept, reject, and assign orders – at time $k$ and $y_k = (y_{k:1}, y_{k:2}, q_k)$ represent an instantaneous state after making those decisions at the beginning of period $k$.

Let us define a set functions, $\Pi \left( y_k; Z_k, W_k, L_k \right) = (y_k^{11}, y_k^{12}, ..., y_k^{l1}, y_k^{l2} W_k^{l1} + Z_k^{l1}, y_k^{l2} W_k^{l2} + Z_k^{l2}, ..., y_k^{ll} W_k^{ll})$, which represent the state after random events – arrivals of new orders, orders abandonment, service completions – occur at the end of period $k$. Then, the expected profit for remaining periods by changing the given state $x_k$ to $y_k$ at the beginning of period $k$ is

$$
\psi_k \left( x_k, y_k \right) = \sum_{i=1}^{l} \left( r_k^i \left( y_{k:1}^i - x_{k:1}^i + \sum_{j=1}^{l} \left( q_k^{ij} - g_k^i \right) \right) - b_{k:1}^i y_{k:1}^i - b_{k:2}^i \mathbb{E}[y_{k:2}^i W_k^i] \right) \\
- \sum_{i=1}^{l} \sum_{j=1}^{l} c_k^{ij} q_k^i + \mathbb{E} \left[ v_{k+1} \left( \Pi \left( y_k; Z_k, W_k, L_k \right) \right) \right]
$$

$$
= - \sum_{i=1}^{l} r_k^i x_{k:1}^i - \sum_{i=1}^{l} \sum_{j=1}^{l} r_k^i g_k^j + \sum_{i=1}^{l} \left( r_k^i - b_{k:1}^i \right) y_{k:1}^i - b_{k:2}^i \mathbb{E}[y_{k:2}^i W_k^i] \\
+ \sum_{i=1}^{l} \sum_{j=1}^{l} \left( r_k^i - c_k^{ij} \right) q_k^i + \mathbb{E} \left[ v_{k+1} \left( \Pi \left( y_k; Z_k, W_k, L_k \right) \right) \right]
$$

Then we can decompose $\psi_k \left( x_k, y_k \right)$ like $\psi_k \left( x_k, y_k \right) = \varphi_k (x_k) + \phi_k (y_k)$, where
\[ \varphi_k(x_k) = - \sum_{i=1}^{I} r_k^i x_{k:1}^i - \sum_{i=1}^{I} \sum_{j=1}^{I} r_k^i g_{k}^{ij} \]

and

\[ \phi_k(y_k) = \sum_{i=1}^{I} \left( (r_k^i - b_{k:1}^i) y_{k:1}^i - b_{k:2}^i \mathbb{E}[y_{k:2}^i W_k^i] \right) + \sum_{i=1}^{I} \sum_{j=1}^{I} (r_k^i - c_{k}^{ij}) q_{k}^{ij} \]

\[ + \mathbb{E} \left[ v_{k+1} \left( \Pi(y_k; Z_k, W_k, L_k) \right) \right] \]

The maximal expected profit given state \( x_k \) at time \( k \), \( v_k(x_k) \), is

\[ v_k(x_k) = \max \psi_k(x_k, y_k) = \max \left( \varphi_k(x_k) + \phi_k(y_k) \right) = \varphi_k(x_k) + \max \phi_y(y_k) \]

Therefore, finding the maximal expected profit is equivalent to the following maximization problem:

\[ \tau_k(x_k) := \max \phi_k(y_k) \]

s.t.

\[ y_{k:1}^i + y_{k:2}^i + \sum_{j=1}^{I} q_{k}^{ij} \leq x_{k:1}^i + x_{k:2}^i + \sum_{j=1}^{I} g_{k}^{ij}, \quad i = 1, 2, \ldots, I \quad (cs1) \]

\[ y_{k:2}^i \leq x_{k:2}^i, \quad i = 1, 2, \ldots, I \quad (cs2) \]

\[ y_{k:1}^i + \sum_{j=1}^{I} q_{k}^{ij} \geq x_{k:1}^i + \sum_{j=1}^{I} g_{k}^{ij}, \quad i = 1, 2, \ldots, I \quad (cs3) \]

\[ \sum_{i=1}^{I} q_{k}^{ij} \leq b_j^i, \quad j = 1, 2, \ldots, J \quad (cs4) \]

\[ q_{k}^{ij} \geq g_{k}^{ij} \quad (cs5) \]

\[ y_{k:1}^i, y_{k:2}^i, q_{k}^{ij} \geq 0 \quad (cs6) \]

And finding the optimal policy is \( y_k(x_k)^* = \arg \max_{y_k} \phi_k(y_k) \) s.t. (cs1) – (cs6).

4 Properties of the Optimal Policy

4.1 Concavity

\( \phi_k(y_k) \) consists of two parts: one is the profit incurred during the current period \( k \), \( \sum_{i=1}^{I} \left( (r_k^i - b_{k:1}^i) y_{k:1}^i - b_{k:2}^i \mathbb{E}[y_{k:2}^i W_k^i] \right) + \sum_{i=1}^{I} \sum_{j=1}^{I} (r_k^i - c_{k}^{ij}) q_{k}^{ij} \), and the other is the profit-to-go part (i.e., the profit expected to incur during the remaining periods), \( \mathbb{E} \left[ v_{k+1} \left( \Pi(y_k; Z_k, W_k, L_k) \right) \right] \). We can easily see
that the first part is linear and so concave on $y_k$. Therefore, in order to check whether $\phi_k(y_k)$ is concave, we only need to check whether the profit-go-to part is concave.

One sufficient condition that the profit-go-to part becomes concave is that $v_{k+1}(y_k)$ is concave because

$$
\Pi \left( y_k; Z_{k}, W_{k}, L_{k} \right) = (y_{k;1}', y_{k;2}', \ldots, y_{k;1}^{l_{k;1}}, y_{k;2}^{l_{k;2}}) + Z_{k+1}^{l_{k+1}}, y_{k;2}^{l_{k;2}} + Z_{k+1}^{2} + \ldots, y_{k;1}^{l_{k;1}} + Z_{k+1}^{l_{k+1}} + q_{k}^{l_{k+1} l_{k}^{1}} + q_{k}^{l_{k+1} l_{k}^{2}} + \ldots, q_{k}^{l_{k+1} l_{k}^{l_{k}}} \right) \) is stochastically linear, which means that given every realization of random variables, $\forall z_{k+1}^{l_{k+1}}, \forall w_{k}^{l_{k}} \in W_{k}^{l_{k}}$, and $\forall l_{k}^{ij} \in L_{k}^{l_{k}}$ for all $i$ and $j$, $\Pi \left( y_k; Z_{k}, W_{k}, L_{k} \right)$ is a linear transformation of $y_k$. Hence, in order to prove the concavity of $\phi_k(y_k)$, now we only need to show that $v_{k+1}(y_k)$ is concave. Theorem 1. shows the concavity of $v_k(y_k)$, for $k = 1, 2, \ldots, n$.

Note that $y_k$ and $x_k$ have the same domain $\mathbb{R}_+^{2l+1}$. Therefore, showing that $v_k(x_k)$ is jointly concave on $x_k$ is equivalent to showing that $v_k(y_k)$ is jointly concave on $y_k$. For simplicity's sake, we use $u_k(x_k)$ instead of $u_k(y_k)$ in Theorem 1.

**Theorem 1.** $u_k(x_k)$ is jointly concave on $x_k$, for $k = 1, 2, \ldots, n$.

**Sketch of the proof of Theorem 1.**

The proof basically shows that $u_k(x_k)$ satisfies the following definition of the concavity:

$$
u_k(\alpha x_a + (1 - \alpha) x_b) \geq \alpha u_k(x_a) + (1 - \alpha)u_k(x_b), \text{ where } x_a \text{ and } x_b \text{ are two arbitrary points in the space of } x_k, \mathbb{R}_+^{2l+1}, \text{ and any } \alpha \in [0,1].$$

Let $y_a^* = \arg\max_{x_k^*} \psi_k(x_a, y_k)$ s.t. (cs1) – (cs6) and $y_b^* = \arg\max_{x_k^*} \psi_k(x_b, y_k)$ s.t. (cs1) – (cs6). Then one sufficient condition for $v_k(x_k)$ to satisfy the above definition is that $\alpha y_a^* + (1 - \alpha)y_b^*$ is a feasible point given $\alpha x_a + (1 - \alpha) x_b$. If $\alpha y_a^* + (1 - \alpha)y_b^*$ is feasible, $u_k(\alpha x_a + (1 - \alpha)x_b) \geq 

\psi_k(\alpha x_a + (1 - \alpha)x_b, \alpha y_a^* + (1 - \alpha)y_b^*)$ since $u_k(x_k) = \max \psi_k(x_k, y_k)$ s.t. (cs1) – (cs6). Also, note that if $\psi_k(x_k, y_k)$ is concave, $\psi_k(\alpha x_a + (1 - \alpha)x_b, \alpha y_a^* + (1 - \alpha)y_b^*) \geq \psi_k(x_a, y_a^*) + \psi_k(x_b, y_b^*) = \alpha \psi_k(x_a) + (1 - \alpha)u_k(x_b)$. As we already showed, if $u_{k+1}(x_k)$ is concave, $\psi_k(x_k, y_k)$ is concave. Therefore, there is a recursive relation. Hence, if $v_{n+1}(x_{n+1})$ is concave, by the principle of mathematical induction, $u_k(x_k)$, for $k = n, n-1, \ldots 1$, become concave as well.

**Proof of Theorem 1.**

Recall that $x_k = (x_{k;1}, x_{k;2}, q_k)$, $y_k = (y_{k;1}, y_{k;2}, q_k)$, and the original problem is

$$
u_k(x_k) := \max \psi_k(x_k, y_k)$$

10
Let $x_a^*$ and $x_b^*$ be two arbitrary points in $\mathbb{R}^{2l+1j}_+$. Let $y_a^*$ and $y_b^*$ be the corresponding optimal solutions of the above maximization problem given $x_a$ and $x_b$, respectively. (i.e., $y_a^* = \text{argmax}_{y_a} \psi_k(x_a, y_a)$ s.t. (cs1) – (cs6) and $y_b^* = \text{argmax}_{y_b} \psi_k(x_b, y_b)$ s.t. (cs1) – (cs6)). Consider two points, $\alpha x_a + (1 - \alpha)x_b$ and $\alpha x_a^* + (1 - \alpha)y_b^*$, for any given $\alpha \in [0,1]$. Then $\alpha y_a^* + (1 - \alpha)y_b^*$ are feasible with respect to constraints (cs1) – (cs6), given $\alpha x_a$ and $(1 - \alpha)x_b$, respectively since if $a \geq b$ and $\alpha \geq 0$, $\alpha a > \alpha b$. Moreover, $\alpha y_a^* + (1 - \alpha)y_b^*$ is feasible with respect to constraints (cs1) – (cs6) given $\alpha x_a + (1 - \alpha)x_b$ since if $a \geq b$ and $c \geq d$, $a + c \geq b + d$:

$$
\alpha y_{a,k:1}^* + (1 - \alpha)y_{b,k:1}^* + \alpha y_{a,k:2}^* + (1 - \alpha)y_{b,k:2}^* + \sum_{j=1}^{l}(\alpha q_{a,k}^{ij} + (1 - \alpha)q_{b,k}^{ij}) \leq \alpha x_{a,k:1} + (1 - \alpha)x_{b,k:1} + \sum_{j=1}^{l}(\alpha q_{a,k}^{ij} + (1 - \alpha)q_{b,k}^{ij}), \quad i = 1, 2, ..., l \quad (cs1)
$$

$$
\alpha y_{a,k:2}^* + (1 - \alpha)y_{b,k:2}^* \leq \alpha x_{a,k:2}^* + (1 - \alpha)x_{b,k:2}^*, \quad i = 1, 2, ..., l \quad (cs2)
$$

$$
\alpha y_{a,k:1}^* + (1 - \alpha)y_{b,k:1}^* + \sum_{j=1}^{l}(\alpha q_{a,k}^{ij} + (1 - \alpha)q_{b,k}^{ij}) \geq 
\alpha x_{a,k:1}^* + (1 - \alpha)x_{b,k:1}^* + \sum_{j=1}^{l}(\alpha g_{a,k}^{ij} + (1 - \alpha)g_{b,k}^{ij}), \quad i = 1, 2, ..., l \quad (cs3)
$$

$$
\sum_{j=1}^{l}(\alpha q_{a,k}^{ij} + (1 - \alpha)q_{b,k}^{ij}) \leq b^j, \quad j = 1, 2, ..., J \quad (cs4)
$$

$$
\sum_{j=1}^{l}(\alpha q_{a,k}^{ij} + (1 - \alpha)q_{b,k}^{ij}) \geq \sum_{j=1}^{l}(\alpha g_{a,k}^{ij} + (1 - \alpha)g_{b,k}^{ij}) \quad (cs5)
$$

If $\psi_k(x_k, y_k)$ is jointly concave on $(x_k, y_k)$ (i.e., $u_{k+1}(y_k)$ is jointly concave on $y_k$), by the definition of concavity, $\psi_k(\alpha x_a + (1 - \alpha)x_b, \alpha y_a^* + (1 - \alpha)y_b^*) \geq \alpha \psi_k(x_a, y_a^*) + (1 - \alpha)\psi_k(x_b, y_b^*) = \alpha u_k(x_a) + (1 - \alpha)u_k(x_b)$. Since $\alpha y_a^* + (1 - \alpha)y_b^*$ satisfies constraints (cs1) – (cs6) given $\alpha x_a + (1 - \alpha)x_b$.
\[ \alpha x_k + (1 - \alpha) y_k = \max \psi_k(\alpha x_n + (1 - \alpha) x_k, y_k) \geq \psi_k(\alpha x_k, \alpha y_k^* + (1 - \alpha) y_k^*) = \alpha u_k(x_k) + (1 - \alpha) u_k(y_k). \]

Therefore, \( u_k(x_k) \) is jointly concave on \( x_k \).

Therefore, if \( u_{n+1}(x_{n+1}) \) is jointly concave (e.g., \( u_{n+1}(x_{n+1}) = 0 \)), by the principle of mathematical induction, \( u_k(x_k) \) is jointly concave on \( x_k \), \( \psi_k(x_k, y_k) \) is jointly concave on \( (x_k, y_k) \), and \( \phi_k(x_k) \) is jointly concave on \( x_k \). Q.E.D.

Given any \( x_k \), the feasible set of \( y_k \) always becomes a closed convex set: Every constraint composes a convex set; therefore, the intersection of all these constraints becomes a closed convex set. \( \phi_k(y_k) \) is jointly concave on \( y_k \). Hence, \( y_k, (x_k)^* \) always exists.

4.2 Submodularity

Now we investigate the submodularity of \( \psi_k(x_k, y_k) \) on \( (x_k, y_k) \). As we already showed, \( \psi_k(x_k, y_k) = \phi_k(x_k) + \phi_k(y_k) \). \( \phi_k(x_k) = -r_{k,1}x_{k,1} - r_{k,2}y_{k,2} + \ldots + r_{k,1}y_{k,1} + \frac{1}{2}g_{k,1}g_{k,1} \); so \( \phi_k(x_k) \) is (sub) modular on \( (x_k, y_k) \). Note that the sum of two submodular functions still becomes submodular. Therefore, if \( \phi_k(y_k) \) is submodular on \( (x_k, y_k) \), \( \psi_k(x_k, y_k) \) becomes submodular on \( (x_k, y_k) \).

Let \( \phi_k(y_k) = \sum_{i=1}^{|\Pi|} \left( (r_{i,1}^k - b_{i,1}^k) y_{i,1}^k - b_{i,2}^k \mathbb{E}[y_{i,2}^k W_{i,2}^k] \right) + \sum_{i=1}^{|\Pi|} \sum_{j=1}^{|\Pi|} (r_{i,2}^k - c_{i,2}^k) q_{i,j}^k u_{i,j}^k \) and \( \phi_k^2(y_k) = \mathbb{E} \left[ u_{k+1} \left( \Pi \left( y_k \mid Z_k, W_k, L_k \right) \right) \right] \). Then \( \phi_k(y_k) = \phi_k^1(y_k) + \phi_k^2(y_k) \). It is also easy to find that \( \phi_k^1(y_k) \) is linear and (sub)modular on \( (x_k, y_k) \).

Therefore, what we need to show for the submodularity of \( \psi_k(x_k, y_k) \) on \( (x_k, y_k) \) is reduced to show that \( \phi_k^2(y_k) \) is submodular on \( y_k \) (or equivalently on \( (x_k, y_k) \)).

Lemma 2. If \( u_{k+1}(y_{k+1}) \) is submodular on \( y_k \), \( \mathbb{E} \left[ u_{k+1} \left( \Pi \left( y_k \mid Z_k, W_k, L_k \right) \right) \right] \) is submodular on \( y_k \).

Proof of Lemma 2.

Recall set function \( \Pi: \mathbb{R}_+^{2^{|\Pi|}+1} \rightarrow \mathbb{R}_+^{2^{|\Pi|}+1} \), such that \( \Pi \left( y_k \mid Z_k, W_k, L_k \right) \) = \( \left( y_{k,1}, y_{k,2}, \ldots, y_{k,|\Pi|}, y_{k,|\Pi|}^1 W_{k,1}^1 + Z_{k,1}, y_{k,|\Pi|}^2 W_{k,2}^2 + Z_{k,2}, \ldots, y_{k,|\Pi|}^{|\Pi|} W_{k,|\Pi|}^{|\Pi|} + Z_{k,|\Pi|}, q_{k,1,1}, q_{k,1,2}, \ldots, q_{k,|\Pi|,|\Pi|} \right) \).

Consider any four points \( y_a, y_b, y_c \) and \( y_d \) in \( \mathbb{R}_+^{2^{|\Pi|}+1} \) such that \( y_c = y_a \lor y_b \) and \( y_d = y_a \land y_b \), which composes a rectangle \( y_a \times y_c \times y_b \times y_d \). Then four points, \( \Pi(y_a), \Pi(y_b), \Pi(y_c), \) and \( \Pi(y_d) \), given any realized values for \( Z_k, W_k, \) and \( L_k \) also satisfies \( \Pi(y_c) = \Pi(y_a) \lor \Pi(y_b) \) and \( \Pi(y_d) = \Pi(y_a) \land \Pi(y_b) \) and
compose a rectangle $\prod(y_a) \prod(y_c) \prod(y_b) \prod(y_d)$. Therefore, for every realization of random variables, 
$Z_k, W_k$, and $L_k, v_k \left( \prod(y_a) \right) + v_k \left( \prod(y_b) \right) \geq v_k \left( \prod(y_c) \right) + v_k \left( \prod(y_d) \right)$; therefore, $E \left[ v_k \left( \prod(y_a) \right) + v_k \left( \prod(y_b) \right) \right] \geq E \left[ v_k \left( \prod(y_c) \right) + v_k \left( \prod(y_d) \right) \right]$. Q.E.D.

Lemma 2 implies that if $v_{k+1}(y_{k+1})$ is submodular on $y_{k+1}$, then $\psi_k(x_k, y_k)$ becomes submodular on $(x_k, y_k)$. Theorem 3 shows that $v_k(y_k)$ is submodular on $y_k$, for $k = 1, 2, \ldots, n$, given $v_{n+1}(y_{n+1})$ is submodular on $y_{n+1}$. For simplicity’s sake, we use $v_k(x_k)$ instead of $v_k(y_k)$.

**Theorem 3.** $v_k(x_k)$ is submodular on $x_k$, for $k = 1, 2, \ldots, n$.

**Sketch of the proof of Theorem 3.**

One way to show a function on a multi-dimensional space is submodular is to show that the function is submodular on every possible two-dimensional subspace of the original space. Hence, in Theorem 3, we basically show that $v_k(x_k)$ is submodular on every possible two-dimensional subspace of the space of $x_k$.

In order to show the submodularity of a multi-dimensional function in a two-dimensional space, we chose two variables (i.e., two dimensions) from the original set of variables and assume that all other variables in other dimensions are fixed (i.e., being constant). For the sake of simplicity, let $\phi_k(x_k)$, $\tau_k(x_k)$, $\psi_k(x_k, y_k)$, $\varphi_k(x_k)$, and $\phi_k(x_k)$ also represent two-dimensional functions projected on the selected two dimensions. For example, $\phi_k(x_{k,1}, x_{k,2}) = \phi_k(x_{k,1}^1 = \theta_{k,1}^1, x_{k,2}^2 = \theta_{k,2}^2, \ldots, x_{k,2}^{i-1} = \theta_{k,2}^{i-1}, x_{k,2}^{i+1} = \theta_{k,2}^{i+1}, \ldots, x_{k,2}^1 = \theta_{k,1}^1, x_{k,1}^2 = \theta_{k,2}^2, \ldots, x_{k,1}^{i-1} = \theta_{k,1}^{i-1}, x_{k,1}^{i+1} = \theta_{k,1}^{i+1}, \ldots, x_{k,1}^1 = \theta_{k,1}^1)$ represent a two-dimensional function which projects the original $\phi_k(x)$ on $(x_{k,1}^1, x_{k,1}^2) \in \mathbb{R}_+^2$ while keeping all other variables remain fixed ($= \theta$).

Recall that $u_k(x_k) = \max \psi_k(x_k, y_k) = \varphi_k(x_k) + \max \phi_k(y_k)$ and $\tau_k(x_k) = \max \phi_k(y_k)$. Since $\varphi_k(x_k)$ is modular (i.e., submodular), if $\tau_k(x_k)$ is submodular, $u_k(x_k)$ becomes submodular. Therefore, Theorem 3 actually shows that $\tau_k(x_k)$ is submodular. In order to show the submodularity of $\tau_k(x_k)$, for any given two-dimensional subspace of the space of $x_k$, $\mathbb{R}_+^{2l+1}$, and any given four points, $x_a, x_b, x_c$ and $x_d$ in that two-dimensional space, which satisfies $x_c = x_a \cup x_b$ and $x_d = x_a \cap x_b$ and compose rectangle $x_a,x_c,x_b,x_d$, we show that $\tau_k(x_a) + \tau_k(x_b) \geq \tau_k(x_c) + \tau_k(x_d)$.

In order to prove Theorem 3, we first need following properties.
Lemma 4. Let \( f(x) \) be a concave function in \( x \). Then, \( f(a) + f(b) \leq f(a + \delta) + f(b - \delta) \), where \( a \leq a + \delta \leq b \) and \( a \leq b - \delta \leq b \).

Proof of Lemma 4.

Since \( f(x) \) is concave,

\[
\frac{\delta}{b-a} f(a) + \left(1 - \frac{\delta}{b-a}\right) f(b) \geq f\left(\frac{\delta}{b-a} a + \left(1 - \frac{\delta}{b-a}\right) b\right) = f(b - \delta)
\]

\[
(1 - \frac{\delta}{b-a}) f(a) + \left(\frac{\delta}{b-a}\right) f(b) \geq f\left(\left(1 - \frac{\delta}{b-a}\right) a + \left(\frac{\delta}{b-a}\right) b\right) = f(a + \delta)
\]

Summing two inequalities results in \( f(a) + f(b) \geq f(a + \delta) + f(a + \delta) \). Q.E.D.

Lemma 5. If \( f(x_1, x_2) \) is jointly concave and submodular on \( (x_1, x_2) \in \mathbb{R}^2_+ \), \( f(x_1, x_2) \) has the following directional properties:

![Diagram](a) and (b)

Figure 4. Directional properties in Lemma 5

Given any four points like the above lozenges where \( a, b, c, d, e, \alpha, \gamma, \varepsilon, \beta, \delta \geq 0 \), \( f(a, b + d) + f(a + c, b + e) \geq f(a, b) + f(a + c, b + d + e) \) and \( f(\alpha, \beta) + f(\alpha + \varepsilon, \beta + \delta) \). Adding three inequalities, (1), (2), and (3), results in \( f(A') + f(B') \geq f(C') + f(D') \).

Proof of Lemma 5.

Case (a): Since function \( f \) is jointly concave, 1) \( f(A') + f(C') \geq f(A) + f(C') \) and 2) \( f(B') + f(D') \geq f(B) + f(D') \). Also, since function \( f \) is submodular, 3) \( f(A) + f(B) \geq f(C) + f(D) \). Adding three inequalities, (1), (2), and (3), results in \( f(A') + f(B') \geq f(C') + f(D') \).

Case (b): Since function \( f \) is jointly concave, 1) \( f(A') + f(D') \geq f(A) + f(D') \) and 2) \( f(B') + f(C') \geq f(B) + f(C') \). Also, since function \( f \) is submodular, 3) \( f(A) + f(B) \geq f(C) + f(D) \). Adding three inequalities, (1), (2), and (3), results in \( f(A') + f(B') \geq f(C') + f(D') \). Q.E.D.
Among variables in $x_k$, we can define eleven types of variable pairs: 1) $x_{k:1}^i$ vs. $x_{k:2}^i$, 2) $x_{k:1}^i$ vs. $g_k^{ij}$, 3) $x_{k:1}^i$ vs. $g_k^{ij}$, 4) $x_{k:1}^i$ vs. $x_{k:2}^i$, 5) $x_{k:1}^i$ vs. $x_{k:2}^i$, 6) $x_{k:1}^i$ vs. $x_{k:2}^i$, 7) $x_{k:1}^i$ vs. $x_{k:2}^i$, 8) $x_{k:1}^i$ vs. $x_{k:2}^i$, 9) $x_{k:1}^i$ vs. $g_k^{ij}$, 10) $g_k^{ij}$ vs. $g_k^{ij}$, and 11) $g_k^{ij}$ vs. $g_k^{ij}$. In order to show that $\tau_k(x_k)$ is submordular, for every pair of variables, we need to establish the following.

Note that $x_k = (x_{k:1}, x_{k:2}, \ldots, x_{k:2}, g_k^{11}, g_k^{12}, \ldots, g_k^{ij}) \in \mathbb{R}_{+}^{2t+j}$. Select any arbitrary two variables, say $x_i$ and $x_j$ ($i \neq j$), in $x_k$ and assume the other variables in $x_k$ are fixed. Consider four points $x_a, x_b, x_c$, and $x_d$ on $(x_i, x_j)$ space like the following figure where $d_j = (d_i, 0)$, $d_i > 0$ and $d_j = (0, d_j)$, $d_j > 0$ so that $x_c = x_a \lor x_b$ and $x_d = x_a \land x_b$.

Figure 6. Four points
Then, the submodularity of $\tau_k(x_k)$ on $(x_i, x_j)$ holds if $\tau_k(x_a) + \tau_k(x_b) \leq \tau_k(x_c) + \tau_k(x_d)$ for all possible such four points, $x_a, x_b, x_c$, and $x_d$ on $(x_i, x_j)$.

Let $y^*_a = \arg\max_{x_k} \phi_k(x_c)$ s.t. (cs1) – (cs6), and $y^*_d = \arg\max_{x_k} \phi_k(x_d)$ s.t. (cs1) – (cs6). In order to prove the submodularity of $\tau_k(x_k)$ on $(x_i, x_j)$, we need to find $y_a$ and $y_b$ which satisfy the following two conditions:

**Con1** $y_a$ and $y_b$ are feasible given $x_a$ and $x_b$, respectively.

**Con2** $\phi_k(y_a) + \phi_k(y_b) \geq \phi_k(y^*_a) + \phi_k(y^*_d) = \tau_k(x_c) + \tau_k(x_d)$

The above two conditions lead $\tau_k(x_a) \geq \phi_k(y_a)$ and $\tau_k(x_b) \geq \phi_k(y_b)$; therefore, $\tau_k(x_a) + \tau_k(x_b) \geq \tau_k(x_c) + \tau_k(x_d)$.

$x_{k:1}^i$ vs. $x_{k:2}^i$ case:

Assume that all other variables are fixed except $x_{k:1}^i$ and $x_{k:2}^i$. Then the following constraints are affected as $x_{k:1}^i$ and $x_{k:2}^i$ changes, while the others are not.

\[
y_{k:1}^i + y_{k:2}^i \leq x_{k:1}^i + x_{k:2}^i + \sum_{j=1}^{J} g_{i}^{ij} - \sum_{j=1}^{J} q_{i}^{ij} \quad \text{(cs1)}
\]

\[
y_{k:2}^i \leq x_{k:2}^i \quad \text{(cs2)}
\]

\[
y_{k:1}^i \geq x_{k:1}^i + \sum_{j=1}^{J} g_{i}^{ij} - \sum_{j=1}^{J} q_{i}^{ij} \quad \text{(cs3)}
\]

\[
y_{k:1}^i, y_{k:2}^i \geq 0 \quad \text{(cs6)}
\]

The feasible areas of $y_a, y_b, y_c$, and $y_d$ given $x_a, x_b, x_c$, and $x_d$ can be characterized as areas a, b, c, and d, respectively in the following figure.

**Figure 7. Feasible areas of $(y_{k:1}^i, y_{k:2}^i)$**

16
Similarly, we can characterize all patterns of feasible areas for all eleven types of variable pairs and how they are changed by adding $d_i$ and/or $d_j$. The following table shows the patterns of feasible areas and changes.

**Table 1. Patterns of feasible areas of the corresponding pair variable in $y_k$**

<table>
<thead>
<tr>
<th>No.</th>
<th>Pattern</th>
<th>Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><img src="image1" alt="Pattern 1" /></td>
<td>$x_1^i$ vs. $x_2^j$ $x_2^j$ vs. $g^{ij}$ $x_1^i$ vs. $g^{ij}$</td>
</tr>
<tr>
<td>2</td>
<td><img src="image2" alt="Pattern 2" /></td>
<td>$x_1^i$ vs. $g^{ij}$</td>
</tr>
<tr>
<td>3</td>
<td><img src="image3" alt="Pattern 3" /></td>
<td>$x_1^i$ vs. $x_2^j$ $x_2^j$ vs. $g^{ij}$ $x_1^i$ vs. $g^{ij}$ $x_1^i$ vs. $g^{ij}$</td>
</tr>
<tr>
<td>4</td>
<td><img src="image4" alt="Pattern 4" /></td>
<td>$x_2^j$ vs. $x_2^j$</td>
</tr>
<tr>
<td>5</td>
<td><img src="image5" alt="Pattern 5" /></td>
<td>$g^{ij}$ vs. $g^{ij}$ $g^{ij}$ vs. $g^{ij}$ $g^{ij}$ vs. $g^{ij}$</td>
</tr>
</tbody>
</table>

Although we construct $x_c$ and $x_d$ such that $x_c < x_d$, it does not guarantee that $y_c^* < y_d^*$. There are nine cases of all possible relative positions of $y_c^*$ and $y_d^*$ in a two dimensional plane. The following figure shows them.
In order to prove that \( \tau_k(x_k) \) is submodular in \( x_k \), we need to find \( y_a \) and \( y_b \) which satisfies the above two conditions (Con 1) and (Con 2) for every pair of variables in \( x_k \) (eleven pairs), every possible four points of rectangular in the selected two-dimensional space, and every possible relative position of \( y_c^* \) and \( y_d^* \) among nine cases in Figure 8, considering the pattern of feasible areas of the corresponding pair of variables in \( y_k \) (see Table 1). Although there are many cases to prove, generalized constructs for \( y_a \) and \( y_b \) can be made. The following table shows the constructs.

**Table 2. Constructs for \( y_a \) and \( y_b \)**

<table>
<thead>
<tr>
<th>Positions of ( y_c^* ) and ( y_d^* )</th>
<th>Constructs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>( y_a = y_b = y_c^* = y_d^* ). Therefore, ( \phi_k(y_a) + \phi_k(y_b) = \phi_k(y_c^<em>) + \phi_k(y_d^</em>) )</td>
</tr>
</tbody>
</table>
| Cases 2, 4, 6, and 7 | The basic idea is we can always find two interior points between \( y_c^* \) and \( y_d^* \) such that \( y_a = y_c^* + \delta \left( y_d^* - y_c^* \right) \) and \( y_b = y_d^* + \delta \left( y_c^* - y_d^* \right) \), where \( 0 \leq \delta \leq 1 \), and \( y_a \) and \( y_b \) are feasible given \( x_a \) and \( x_b \) (i.e., they satisfy CON 1). Then due to the concavity of \( \phi_k(y_k) \), by Lemma 4, \( \phi_k(y_a) + \phi_k(y_b) \geq \phi_k(y_c^*) + \phi_k(y_d^*) \) (i.e., they satisfy CON 2) in any pattern of feasible areas of the corresponding pair variable in \( y_k \).

We find minimal \( \delta_1 \) (0 \leq \( \delta_1 \) \leq 1) such that \( y_d^* + \delta_1 \left( y_c^* - y_d^* \right) \) becomes feasible given \( x_b \) (or \( x_a \)) and minimal \( \delta_2 \) (0 \leq \( \delta_2 \) \leq 1) such that \( y_c^* + \delta_2 \left( y_d^* - y_c^* \right) \) becomes feasible given \( x_a \) (or \( x_b \)), respectively. Let \( \delta = \max(\delta_1, \delta_2) \). Let \( y_a = y_c^* + \delta \left( y_d^* - y_c^* \right) \) and \( y_b = y_d^* + \delta \left( y_c^* - y_d^* \right) \) for every pair of variables in \( x_k \). Then due to the concavity of \( \phi_k(y_k) \), by Lemma 4, \( \phi_k(y_a) + \phi_k(y_b) \geq \phi_k(y_c^*) + \phi_k(y_d^*) \) (i.e., they satisfy CON 2) in any pattern of feasible areas of the corresponding pair variable in \( y_k \).
\( y^*_a \) and \( y^*_b = y^*_a + \delta \left( y^*_c - y^*_d \right) \). Then \( y^*_a \) and \( y^*_b \) are feasible given \( x_a \) and \( x_b \), respectively. Due to the concavity of \( \phi_k \left( y_k \right) \), by Lemma 4, \( \phi_k \left( y^*_a \right) + \phi_k \left( y^*_b \right) \geq \phi_k \left( y^*_c \right) + \phi_k \left( y^*_d \right) \) holds.

Cases 3, 5, and 8

These cases do not happen in any pattern of feasible areas of the corresponding pair variable in \( y_k \).

Case 9

In this case, there are three sub-cases to construct \( y^*_a \) and \( y^*_b \). First, if \( y^*_{d,j} - y^*_{c,j} > d_j \), \( y^*_a = y^*_{c} + d_j \) and \( y^*_b = y^*_{d} - d_j \). Second, if \( y^*_{d,i} - y^*_{c,i} > d_i \), \( y^*_b = y^*_{c} + d_j \) and \( y^*_a = y^*_{d} - d_j \). Lastly, if \( y^*_{d,j} - y^*_{c,j} \leq d_j \) and \( y^*_{d,i} - y^*_{c,i} \leq d_i \), \( y^*_a = \left( y^*_{c,j}, y^*_{d,j} \right) \) and \( y^*_b = \left( y^*_{d,i}, y^*_{d,i} \right) \). The above figures (a), (b), and (c) correspond to each sub-case, respectively.

We can easily find that \( y^*_a \) and \( y^*_b \) in any sub-case are feasible in any pattern of feasible area, given \( x_a \) and \( x_b \), respectively. Note that if \( y^*_{d,j} - y^*_{c,j} > d_j \) and \( y^*_{d,i} - y^*_{c,i} > d_i \), either (a) or (b) is fine.

We already showed that if \( v_{k+1} \left( y_{k+1} \right) \) is submodular, \( \phi_k \left( y_k \right) \) becomes submodular. If \( \phi_k \left( y_k \right) \) is submodular, \( y^*_a, y^*_b, y^*_c, \) and \( y^*_d \) in sub-case (c) satisfies \( \phi_k \left( y^*_a \right) + \phi_k \left( y^*_b \right) \geq \phi_k \left( y^*_c \right) + \phi_k \left( y^*_d \right) \). Therefore, if \( v_{n+1} \left( y_{n+1} \right) \) is assumed to be submodular (e.g. 0 or a linear function), by the mathematical induction, the submodularity condition holds in sub-case (c).

We already showed that \( \phi_k \left( y_k \right) \) is jointly concave in Theorem 1 and that if \( v_{k+1} \left( y_{k+1} \right) \) is submodular, \( \phi_k \left( y_k \right) \) becomes submodular in Lemma 2. If \( \phi_k \left( y_k \right) \) is submodular and jointly concave, by Lemma 5, \( y^*_a, y^*_b, y^*_c, \) and \( y^*_d \) in sub-cases (a) and (b) always satisfy \( \phi_k \left( y^*_a \right) + \phi_k \left( y^*_b \right) \geq \phi_k \left( y^*_c \right) + \phi_k \left( y^*_d \right) \). Therefore, if \( v_{n+1} \left( y_{n+1} \right) \) is assumed to be submodular (e.g. 0 or a linear function), by the mathematical induction, the submodularity condition holds in sub-cases (a) and (b).
The above constructs find that \( \tau_k(x_k) \) is submodular. Hence, \( u_k(x_k) \) is submodular. \textbf{Q.E.D.}

### 4.3 Directional Properties

The properties about concavity and submodularity which we already found provide useful information about the structural properties of the optimal policy. However, in our model, some decisions are associated with moving along with the line of -45 degree (i.e., \((1, -1)\) direction) between two variables, which the properties of concavity and submodularity cannot tell much about. In order to find additional structural properties of the optimal policy regarding these decisions, we investigate the directional properties of \( \psi_k(x_k, y_k), \phi_k(y_k), \) and \( u_k(y_k) \). We are only interested in three pairs of variables, \((x_{k:1}^i, x_{k:2}^i), (g_{k}^{ij}, x_{k:1}^i), \) and \((g_{k}^{ij}, x_{k:2}^i)\) since only they are associated with the movements along with this direction: \textit{accept} \((x_{k:2}^i \rightarrow x_{k:1}^i)\), \textit{assign} \((x_{k:1}^i \rightarrow g_{k}^{ij})\), and \textit{accept and assign} \((x_{k:2}^i \rightarrow g_{k}^{ij})\) decisions.

Consider four points which compose lozenges like the following figures. Let lozenge (i) consist of four feasible points in \( \mathbb{R}_+^2: A = (a, b); C = (a, d); B = (a + \delta, d - \delta); D = (a + \delta, b - \delta) \), where \( b > d > \delta > 0 \) and lozenge (ii) consists of four feasible points in \( \mathbb{R}_+^2: A = (a, b); D = (c, b); B = (c + \delta, b - \delta); C = (a + \delta, b - \delta) \) where \( c > a > 0 \) and \( b > \delta > 0 \).

![Figure 9. Directional properties along with -45 degree line](image)

Let \( f(x) \) be a function defined in \( \mathbb{R}_+^2 \), where \( x = (x_i, x_j) \). Similar to the definition of submodularity and Lemma 5, we can also compare the values of \( f(A) + f(B) \) and \( f(C) + f(D) \). This analysis provides the properties of function, \( \tilde{f}^*(m) = \max f(x) \) s.t. \( x_i + x_j = m \) and the points, \((\tilde{x}_i^*(m), \tilde{x}_j^*(m)) = \arg\max_{(x_i, x_j)} f(x) \) s.t. \( x_i + x_j = m \).

In order to show some directional properties, we need following properties about concavity.
Lemma 6. Let \( f(x) \) be a jointly concave function in \( \mathbb{R}^2 \), where \( x = (x_i, x_j) \). Let \( f^*_{x_i}(x) = \max f(x) \) s.t. \( x_i = x \) and \( x^*_j(x) = \arg \max_{x_j} f(x) \) s.t. \( x_i = x \). Similarly, \( f^*_{x_j}(x) = \max f(x) \) s.t. \( x_j = x \) and \( x^*_i(x) = \arg \max_{x_i} f(x) \) s.t. \( x_j = x \). Then \( f^*_{x_i}(x) \) and \( f^*_{x_j}(x) \) are concave in \( x \).

Proof of Lemma 6.

Let \( a \) and \( b \) be two arbitrary points in \( x \), \( (a, x^*_j(a)) \). Then we can define three points in \( x \):

\[
(a, x^*_j(a)), (b, x^*_j(b)), \text{ and } (aa + (1 - \alpha)b, \alpha x^*_j(a) + (1 - \alpha)x^*_j(b)), \text{ where } 0 \leq \alpha \leq 1.
\]

Because \( f(x) \) is jointly concave, \( (af(a, x^*_j(a)) + (1 - \alpha)f(b, x^*_j(b)) \leq f(aa + (1 - \alpha)b, \alpha x^*_j(a) + (1 - \alpha)x^*_j(b)) \).

By the definition of \( f^*_{x_j}(x) \), \( f(aa + (1 - \alpha)b, \alpha x^*_j(a) + (1 - \alpha)x^*_j(b)) \leq f^*_{x_j}(aa + (1 - \alpha)b) \).

Therefore, \( \alpha f^*_{x_i}(a) + (1 - \alpha)f^*_{x_j}(b) \leq f^*_{x_i}(aa + (1 - \alpha)b) \). Hence, \( f^*_{x_i}(x) \) is concave in \( x \). The concavity of \( f^*_{x_j}(x) \) can be shown similarly. Q.E.D.

Lemma 7. Let \( f(x) \) be a jointly concave function in \( \mathbb{R}^2 \), where \( x = (x_i, x_j) \). Then \( \tilde{f}^*(m) \) is concave in \( m \), where \( m = x_i + x_j \).

Proof of Lemma 7.

Let \( a \) and \( b \) be two arbitrary points in \( m \). Then we can define three points in \( x \):

\[
(x^*_i(a), x^*_j(a)), (x^*_i(b), x^*_j(b)), \text{ and } (a\tilde{x}^*_i(a) + (1 - \alpha)\tilde{x}^*_j(b), \alpha\tilde{x}^*_i(a) + (1 - \alpha)\tilde{x}^*_j(b)) \text{ where } 0 \leq \alpha \leq 1.
\]

Because \( f(x) \) is jointly concave, \( (af(x^*_i(a), x^*_j(a)) + (1 - \alpha)f(x^*_i(b), x^*_j(b)) \leq f(a\tilde{x}^*_i(a) + (1 - \alpha)\tilde{x}^*_j(b), \alpha\tilde{x}^*_i(a) + (1 - \alpha)\tilde{x}^*_j(b)) \).

By the definition of \( \tilde{f}^*(m) \), \( \tilde{f}^*(a) = f(x^*_i(a), x^*_j(a)), \tilde{f}^*(b) = f(x^*_i(b), x^*_j(b)) \) and \( f(a\tilde{x}^*_i(a) + (1 - \alpha)\tilde{x}^*_j(b), \alpha\tilde{x}^*_i(a) + (1 - \alpha)\tilde{x}^*_j(b)) \leq f^*(a\tilde{x}^*_i(a) + (1 - \alpha)\tilde{x}^*_j(b) + \alpha\tilde{x}^*_j(a) + (1 - \alpha)\tilde{x}^*_i(b)) \).

Note that \( \alpha\tilde{x}^*_i(a) + (1 - \alpha)\tilde{x}^*_i(b) + \alpha\tilde{x}^*_j(a) + (1 - \alpha)\tilde{x}^*_j(b) = \alpha(\tilde{x}^*_i(a) + \tilde{x}^*_j(a)) + (1 - \alpha)(\tilde{x}^*_i(b) + \tilde{x}^*_j(b)) = aa + (1 - \alpha)b \).

Hence, \( \alpha\tilde{f}^*(a) + (1 - \alpha)\tilde{f}^*(b) \leq \tilde{f}^*(aa + (1 - \alpha)b) \).

Therefore, \( \tilde{f}^*(m) \) is concave in \( m \). Q.E.D.

Some directional properties are preserved after the transformation of \( \Pi(\pi, W, Z, Y) \). The following property plays a critical role in finding out the directional properties of our value function.

Lemma 8. For \( (x_{k+1}^{i+1}, x_{k+2}^{j+1}) \) and \( (s_{k+1}^{ij}, x_{k+1}^{j+1}) \), any four points which compose lozenge (i) in Figure 9 satisfy \( f_k(A) + f_k(B) \geq f_k(C) + f_k(D) \) if \( v_{k+1}(A) + u_{k+1}(B) \geq v_{k+1}(C) + u_{k+1}(D) \). Similarly, any four points which compose lozenge (ii) in Figure 9 satisfy \( f_k(A) + f_k(B) \leq f_k(C) + f_k(D) \) if \( v_{k+1}(A) + u_{k+1}(B) \leq v_{k+1}(C) + u_{k+1}(D) \).

Proof of Lemma 8.
For the simplicity’s sake, we use $x_k$ instead of $y_k$. Recall that $\phi_k(x_k) = \phi_k^1(x_k) + \phi_k^2(x_k)$. 

$\phi_k^1(x_k)$ is modular which implies that $\phi_k^1(A) + \phi_k^1(B) = \phi_k^1(C) + \phi_k^1(D)$ in both lozenges. Hence, the directional property of $\phi_k(x_k)$ is determined only by $\phi_k^2(x_k) = \mathbb{E}[u_{k+1}(\Pi(x_k; W_k, Z_k, L_k))] = \mathbb{E}[u_{k+1}(x_{k:1}^1, x_{k:2}^1, \ldots, x_{k:1}^k, x_{k:2}^k W_k^1 + Z_{k+1}^1, x_{k:1}^2, W_k^2 + Z_{k+1}^2, \ldots, x_{k:2}^k W_k^l + Z_{k+1}^l, g_{k:1}^1 L_{k}^1, g_{k:2}^1 L_{k}^2, \ldots, g_{k:2}^1 L_{k}^l)]$.

Like other functions, for the sake of simplicity of notations, let $\Pi(x_{k:1}^i, x_{k:2}^i; W_k^i, Z_{k+1}^i)$ and $\Pi(g_{k:1}^i, x_{k:2}^i; L_{k}^i)$ be the two-dimensional projected functions where only $(x_{k:1}^i, x_{k:2}^i)$ and $(g_{k:1}^i, x_{k:1}^i)$ are variables, respectively while the other variables are remain fixed. The following figure show that how A, B, C, and D in lozenges (i) and (ii) are transformed by $\Pi(x_{k:1}^i, x_{k:2}^i; W_k^i, Z_{k+1}^i)$ and $\Pi(g_{k:1}^i, x_{k:1}^i; L_{k}^i)$ and show new four points, $\Pi(A), \Pi(B), \Pi(C)$, and $\Pi(D)$, moved by these functions.

![Diagram](image.png)

**Figure 10. Transformation by $\Pi(x_{k:1}^i, x_{k:2}^i; W_k^i, Z_{k+1}^i)$ and $\Pi(g_{k:1}^i, x_{k:1}^i; L_{k}^i)$**

Transformed lozenge (i) (i.e., lower lozenge in (i)) in Figure 10 can be decomposed into lozenge ACB’D’ and rectangle D’B’BD like Figure 11 (i). By Proposition 9, $v_{k+1}(A) + v_{k+1}(B') \geq v_{k+1}(C) + v_{k+1}(D')$, and by Theorem 3, $v_{k+1}(D') + v_{k+1}(B) \geq v_{k+1}(B') + v_{k+1}(D)$. By summing two inequalities, $v_{k+1}(A) + v_{k+1}(B) \geq v_{k+1}(C) + v_{k+1}(D)$. Similarly, Transformed lozenge (ii) (i.e., lower lozenge in (ii)) in Figure 10 can be decomposed into lozenge ACB’D’ and rectangle D’B’BD like Figure 11 (ii). By Proposition 9, $v_{k+1}(C) + v_{k+1}(D') \geq v_{k+1}(A') + v_{k+1}(B)$, and by Theorem 3, $v_{k+1}(A') + v_{k+1}(D) \geq v_{k+1}(A) + v_{k+1}(D')$. By summing two inequalities, $v_{k+1}(C) + v_{k+1}(D) \geq v_{k+1}(A) + v_{k+1}(B)$.
As we showed, transformed lozenge (i) always satisfies \( u_{k+1}(\Pi(A)) + u_{k+1}(\Pi(B)) \geq u_{k+1}(\Pi(C)) + u_{k+1}(\Pi(D)) \) and transformed lozenge (ii) always satisfies \( u_{k+1}(\Pi(A)) + u_{k+1}(\Pi(B)) \leq u_{k+1}(\Pi(C)) + u_{k+1}(\Pi(D)) \) for any realization of random variable, \( W_k^i, Z_k^{i+1}, \) and \( L_k^{ij} \). In summary, for lozenge (i), \( \mathbb{E}[u_{k+1}(\Pi(A))] + \mathbb{E}[u_{k+1}(\Pi(B))] \geq \mathbb{E}[u_{k+1}(\Pi(C))] + \mathbb{E}[u_{k+1}(\Pi(D))] \); therefore, \( \phi_k(A) + \phi_k(B) \geq \phi_k(C) + \phi_k(D) \) hold. For lozenge (ii), \( \mathbb{E}[u_{k+1}(\Pi(A))] + \mathbb{E}[u_{k+1}(\Pi(B))] \leq \mathbb{E}[u_{k+1}(\Pi(C))] + \mathbb{E}[u_{k+1}(\Pi(D))] \); therefore, \( \phi_k(A) + \phi_k(B) \leq \phi_k(C) + \phi_k(D) \) hold. Q.E.D.

We show and prove the directional properties while introducing the structure of the optimal policy. In other words, we claim directional properties. The claimed properties should lead a structure of the optimal policy, under which in turn, we show the claimed directional properties hold.

### 4.3.1 Directional Property 1: \((x_{k:1}, x_{k:2})\)

In this section, we only consider \( x_{k:1} \) and \( x_{k:2} \) as variables and assume that all other variables remain fixed. Let \( (\bar{x}_{k:1}, \bar{x}_{k:2}) = \arg \max_{x_{k:1}, x_{k:2}} \phi_k(\bar{x}_{k:1}, \bar{x}_{k:2}) \) s.t. \( x_{k:1} = x_{k:2} \), \( x_{k:2} = x_{k:1} \), and \( x_{k:1}^* = x_{k:2}^* = x_{k:1} \): Let \( \bar{x}_{k:1} = \max_{x_{k:1}} \phi_k(\bar{x}_{k:1}, \bar{x}_{k:2}) \) s.t. \( \bar{x}_{k:2} = \bar{x}_{k:1} \), \( \bar{x}_{k:2} = \max_{x_{k:2}} \phi_k(\bar{x}_{k:1}, \bar{x}_{k:2}) \) s.t. \( \bar{x}_{k:1} = \bar{x}_{k:2} \) and \( \phi_{k,x_{k:1}} = \max_{x_{k:1}} \phi_k(\bar{x}_{k:1}, \bar{x}_{k:2}) \) s.t. \( \bar{x}_{k:1} = x_{k:1} \).

The following figure represents the structure of the optimal policy (i.e., left figure) and the directional properties in \((x_{k:1}, x_{k:2})\) two dimensional space. Lozenges (i) and (ii) presented in the figure are same as those in Figure 9.
**Proposition 9.** For \( k = 1, 2, \ldots, n \), if \( \phi_k(x_k) \) satisfies \( \phi_k(A) + \phi_k(B) \geq \phi_k(C) + \phi_k(D) \) in the case of lozenge (i) and \( \phi_k(A) + \phi_k(B) \leq \phi_k(C) + \phi_k(D) \) in the case of lozenge (ii), then 1) the optimal policy in \( (x_{k:1}, x_{k:2}) \) plane looks like the structure represented in Figure 10 and 2) \( v_k(x) \) satisfies the condition \( v_k(A) + v_k(B) \geq v_k(C) + v_k(D) \) in the case of lozenge (i) and \( v_k(A) + v_k(B) \geq v_k(C) + v_k(D) \) in the case of lozenge (ii).

**Proof of Proposition 9.**

We already showed the concavity and submodularity of \( \phi_k(x_k) \) in Theorems 1 and 3. It is trivial to show that the concavity and submodularity as well as the claimed directional properties in cases (i) and (ii) of Figure 10 lead the structure of the optimal policy in \( (x_{k:1}, x_{k:2}) \). First, the concavity implies the existence of the optimal solution, and next the submodularity indicates that \( x_{k:2}^i \) is decreasing in \( x_{k:1}^i \). Lastly, directional properties imply that the decreasing rate of \( x_{k:2}^i \) is steeper than -1 (i.e., 45 degree), \( x_{k:1}^i + x_{k:2}^i \) is decreasing in \( x_{k:1}^i + x_{k:2}^i \), and \( x_{k:2}^i \) is increasing in \( x_{k:1}^i + x_{k:2}^i \). All of these finally result in the structure of the optimal policy shown in Figure 10.

Recall that \( v_k(x_k) = \varphi_k(x_k) + \tau_k(x_k) \). \( \varphi_k(x) \) is modular which means that \( \varphi_k(A) + \varphi_k(B) = \varphi_k(C) + \varphi_k(D) \) in both cases of lozenges (i) and (ii). Therefore, in order to prove that \( v_k(x) \) satisfy the condition \( v_k(A) + v_k(B) \geq v_k(C) + v_k(D) \) in the case of lozenge (i) and \( v_k(A) + v_k(B) \geq v_k(C) + v_k(D) \) in the case of lozenge (ii), we only need to show that \( \tau_k(x) \) satisfy the condition \( \tau_k(A) + \tau_k(B) \geq \tau_k(C) + \tau_k(D) \) in lozenge (i) case and \( \tau_k(A) + \tau_k(B) \geq \tau_k(C) + \tau_k(D) \) in lozenge (ii) case. Let \( \tau_k(x) = \max \phi_k(x_k) \) using the structure of proposed optimal policy.
1. Case of lozenge (i)

1.1. All points in one area

1.1.1. If all four points are in area 1, \( \tau_k(A) = \tau_k(D) \) and \( \tau_k(B) = \tau_k(C) \).

1.1.2. If all four points are in area 2, \( \tau_k(A) = \tau_k(D) = \tau_k(B) = \tau_k(C) \).

1.1.3. If all four points are in area 3, \( \tau_k(A) = \tau_k(C) \) and \( \tau_k(B) = \tau_k(D) \).

1.1.1. If all four points are in area 4, \( \phi_k(x_k) \) satisfies \( \phi_k(A) + \phi_k(B) \geq \phi_k(C) + \phi_k(D) \) and \( \tau_k(x) = \phi_k(x_k) \).

1.2. All points are in two areas:

1.2.1. Areas 1 and 2

1.2.1.1. If \( C \) and \( B \) are in area 1 and \( A \) and \( D \) are in area 2, \( \tau_k(A) = \tau_k(D) = \tau_k(B) = \tau_k(C) \).

1.2.2. Areas 2 and 3

1.2.2.1. If \( A \) and \( C \) are in area 2 and \( B \) and \( D \) are in area 3, \( \tau_k(A) = \tau_k(C) \) and \( \tau_k(B) = \tau_k(D) \).

1.2.3. Areas 3 and 4

1.2.3.1. If \( C \) is in area 4 and \( A, B \) and \( D \) are in area 3, \( \tau_k(A) \geq \tau_k(C) \) and \( \tau_k(B) = \tau_k(D) \).

1.2.3.2. If \( A \) and \( C \) are in area 4 and \( B \) and \( D \) are in area 3, \( \tau_k(A) \geq \tau_k(C) \) and \( \tau_k(B) = \tau_k(D) \).

1.2.3.3. If \( B \) and \( C \) are in area 4 and \( A \) and \( D \) are in area 3, let \( \bar{A} = (a, d + x_2^+(a + \delta) - d) \) and \( \bar{B} = (a + \delta, x_2^+(a + \delta)) \). \( \bar{A}, B, \) and \( \bar{B} \) are all in area 4, \( \tau_k(A) \geq \tau_k(\bar{A}) \) and \( \tau_k(D) \geq \tau_k(\bar{B}) \).

1.2.3.4. If \( A, B, \) and \( C \) are in area 4 and \( D \) is in area 3, let \( \bar{A} = (a, d + x_2^+(a + \delta) - d) \) and \( \bar{B} = (a + \delta, x_2^+(a + \delta)) \). \( \bar{A}, B, \) and \( \bar{B} \) are all in area 4, \( \tau_k(A) \geq \tau_k(\bar{A}) \) and \( \tau_k(D) \geq \tau_k(\bar{B}) \).

1.2.4. Areas 4 and 1

1.2.4.1. If \( D \) is in area 4 and \( A, B \) and \( C \) are in area 1, \( \tau_k(A) \geq \tau_k(D) \) and \( \tau_k(B) = \tau_k(C) \).

1.2.4.2. If \( A \) and \( D \) are in area 4 and \( B \) and \( C \) are in area 1, \( \tau_k(A) \geq \tau_k(D) \) and \( \tau_k(B) = \tau_k(C) \).

1.2.4.3. If \( B \) and \( D \) are in area 4 and \( A \) and \( C \) are in area 1, let \( \tilde{C} = (\tilde{x}_1^+(a, d), \tilde{x}_2^+(a, d)) \) and \( \tilde{A} = (\tilde{x}_1^+(a, d), x_2^+(a, d) + \delta) \). \( \tilde{A}, B, \tilde{C}, \) and \( D \) are all in area 4, \( \tau_k(A) \geq \tau_k(\tilde{A}) \) and \( \tau_k(C) = \tau_k(\tilde{C}) \).

1.2.4.4. If \( C \) is in area 1 and \( A, B \) and \( D \) are in area 4, let \( \tilde{C} = (\tilde{x}_1^+(a, d), \tilde{x}_2^+(a, d)) \) and \( \tilde{A} = (\tilde{x}_1^+(a, d), x_2^+(a, d) + \delta) \). \( \tilde{A}, B, \tilde{C}, \) and \( D \) are all in area 4, \( \tau_k(A) \geq \tau_k(\tilde{A}) \) and \( \tau_k(C) = \tau_k(\tilde{C}) \).

1.2.5. Areas 1 and 3

1.2.5.1. If \( B \) and \( D \) are in area 3 and \( A \) and \( C \) are in area 1, \( \tau_k(A) \geq \tau_k(C) \) and \( \tau_k(B) = \tau_k(D) \).
1.3. All points are in three areas:

1.3.1. Areas 1, 2, and 3

1.3.1.1. If C is in area 1, A is in area 2, and B and D are in area 3, \( \tau_k(A) \geq \tau_k(C) \) and \( \tau_k(B) = \tau_k(D) \).

1.3.2. Areas 1, 3, and 4

1.3.2.1. If C and A are in area 1, B is in area 4, and D is in area 3, let \( \tilde{C} = (\tilde{x}_1^i(a, d), \tilde{x}_2^i(a, d)) \), \( \tilde{D} = (a + \delta, x_2^i(a + \delta)) \), and \( \tilde{A} = (\tilde{x}_1^i(a, d), \tilde{x}_2^i(a, d) + x_2^i(a + \delta) - d) \). Since \( \tilde{A}, B, \tilde{C}, \) and \( \tilde{D} \) are all in area 4, \( \tau_k(A) \geq \tau_k(\tilde{A}) \), \( \tau_k(C) = \tau_k(\tilde{C}) \), and \( \tau_k(D) = \tau_k(\tilde{D}) \).

1.3.2.2. If C and A are in area 1, A and B are in area 4, and D is in area 3, let \( \tilde{C} = (\tilde{x}_1^i(a, d), \tilde{x}_2^i(a, d)) \), \( \tilde{D} = (a + \delta, x_2^i(a + \delta)) \), and \( \tilde{A} = (\tilde{x}_1^i(a, d), \tilde{x}_2^i(a, d) + x_2^i(a + \delta) - d) \). Since \( \tilde{A}, B, \tilde{C}, \) and \( \tilde{D} \) are all in area 4, \( \tau_k(A) \geq \tau_k(\tilde{A}) \), \( \tau_k(C) = \tau_k(\tilde{C}) \), and \( \tau_k(D) = \tau_k(\tilde{D}) \).

1.4. All points are in four areas:

1.4.1.1. If A is in area 2, C is in area 1, B is in area 4, and D is in area 3, let \( \tilde{C} = (\tilde{x}_1^i(a, d), \tilde{x}_2^i(a, d)) \), \( \tilde{D} = (a + \delta, x_2^i(a + \delta)) \), and \( \tilde{A} = (\tilde{x}_1^i(a, d), \tilde{x}_2^i(a, d) + x_2^i(a + \delta) - d) \). Since \( \tilde{A}, B, \tilde{C}, \) and \( \tilde{D} \) are all in area 4, \( \tau_k(A) \geq \tau_k(\tilde{A}) \), \( \tau_k(C) = \tau_k(\tilde{C}) \), and \( \tau_k(D) = \tau_k(\tilde{D}) \).

2. Case of lozenge (ii)

2.1. All points are in one area:

2.1.1. If all four points are in area 1, \( \tau_k(A) = \tau_k(C) \) and \( \tau_k(B) = \tau_k(D) \).

2.1.2. If all four points are in area 2, \( \tau_k(A) = \tau_k(D) \) and \( \tau_k(B) = \tau_k(C) \).

2.1.3. If all four points are in area 3, by Lemmas 6 and 2, \( \tau_k(C) + \tau_k(D) \geq \tau_k(A) + \tau_k(B) \).

2.1.4. If all four points are in area 4, \( \phi_k(x_k) \) satisfies \( \phi_k(C) + \phi_k(D) \geq \phi_k(A) + \phi_k(B) \) and \( \tau_k(x) = \phi_k(x_k) \).

2.2. All points are in two areas

2.2.1. Areas 1 and 2

2.2.1.1. If A and C are in area 1 and B and D are in area 2, \( \tau_k(A) = \tau_k(C) \) and \( \tau_k(B) = \tau_k(D) \).

2.2.2. Areas 2 and 3

2.2.2.1. If A is in area 2 and B, C, and D are in area 3, \( \tau_k(A) = \phi_{k,x_2^i}(x_1^i) \), \( \tau_k(B) = \phi_{k,x_2^i}(c + \delta) \), \( \tau_k(C) = \phi_{k,x_2^i}(a + \delta) \), and \( \tau_k(D) = \phi_{k,x_2^i}(c) \). If \( c \leq a + \delta, c - x_1^i \leq c - a \). Otherwise,
2.2.2. If A and C are in area 2 and B and D are in area 3, \( \tau_k(A) = \tau_k(C) \) and \( \tau_k(B) = \tau_k(D) \).
2.2.2.3. If A and D are in area 2 and B and C are in area 3, \( \tau_k(A) = \tau_k(C) \) and \( \tau_k(B) \leq \tau_k(D) \).
2.2.2.4. If A, C, and D are in area 2 and B is in area 3, \( \tau_k(A) = \tau_k(C) \) and \( \tau_k(B) \leq \tau_k(D) \).

2.2.3. Areas 3 and 4

2.2.3.1. If A is in area 4 and B, C, and D are in area 3, by Lemmas 6 and 2, \( \phi_{k,x_2}^*(a) + \phi_{k,x_2}^*(c + \delta) \leq \phi_{k,x_2}^*(a + \delta) + \phi_{k,x_2}^*(c) \). Due to the property of lozenge (i), \( \tau_k(A) - \phi_{k,x_2}^*(a) \leq \tau_k(C) - \phi_{k,x_2}^*(a + \delta) \leq 0 \). \( \tau_k(B) = \phi_{k,x_2}^*(c + \delta) \), \( \tau_k(D) = \phi_{k,x_2}^*(a + \delta) \).

2.2.3.2. If A and C are in area 4 and B and D are in area 3, by Lemmas 6 and 2, \( \phi_{k,x_2}^*(a) + \phi_{k,x_2}^*(c + \delta) \leq \phi_{k,x_2}^*(a + \delta) + \phi_{k,x_2}^*(c) \). Due to the property of lozenge (i), \( \tau_k(A) - \phi_{k,x_2}^*(a) \leq \tau_k(C) - \phi_{k,x_2}^*(a + \delta) \leq 0 \). \( \tau_k(B) = \phi_{k,x_2}^*(c + \delta) \), \( \tau_k(D) = \phi_{k,x_2}^*(c) \).

2.2.3.3. If A and D are in area 4 and B and C are in area 3, by Lemmas 6 and 2, \( \phi_{k,x_2}^*(a) + \phi_{k,x_2}^*(c + \delta) \leq \phi_{k,x_2}^*(a + \delta) + \phi_{k,x_2}^*(c) \). Due to the property of lozenge (i), \( \tau_k(A) - \phi_{k,x_2}^*(a) \leq \tau_k(C) - \phi_{k,x_2}^*(a + \delta) \leq 0 \). \( \tau_k(B) = \phi_{k,x_2}^*(c + \delta) \), \( \tau_k(D) = \phi_{k,x_2}^*(c) \).

2.2.3.4. If A, C, and D are in area 4 and B is in area 3, draw line \( \ell \) which is parallel to lines AC and BD and passes through point \( \tilde{B} \). Then there are two different cases.

![Diagram](image)

(a) A and C are below \( \ell \)  
(b) A and C are above \( \ell \)

**Figure 13. Directional property 1 – Proof 2.2.3.4**

Figure 13 (a) represents the case when points A and C are below line \( \ell \), and figure (b) represents the case when points A and C are above line \( \ell \).
Case (a): Due to the concavity, by Lemma 2, \( \tau_k(\bar{A}) + \tau_k(\bar{B}) \leq \tau_k(\bar{C}) + \tau_k(\bar{D}) \). Also, \( \tau_k(\bar{A}) - \tau_k(\bar{A}) \geq \tau_k(\bar{D}) - \tau_k(\bar{D}) \), \( \tau_k(\bar{B}) = \tau_k(\bar{B}) \), and \( \tau_k(\bar{C}) \leq \tau_k(\bar{C}) \). Therefore, \( \tau_k(\bar{A}) + \tau_k(\bar{B}) \leq \tau_k(\bar{C}) + \tau_k(\bar{D}) \).

Case (b): Draw another line \( \bar{A}\bar{B} \). Due to the concavity, by Lemma 2, \( \tau_k(\bar{A}) + \tau_k(\bar{B}) \leq \tau_k(\bar{C}) + \tau_k(\bar{D}) \). Also, \( \tau_k(\bar{B}) = \tau_k(\bar{B}) \), \( \tau_k(\bar{C}) \geq \tau_k(\bar{C}) \), and \( \tau_k(\bar{D}) \geq \tau_k(\bar{D}) \). Therefore, \( \tau_k(\bar{A}) + \tau_k(\bar{B}) \leq \tau_k(\bar{C}) + \tau_k(\bar{D}) \).

2.2.4. Areas 1 and 4

2.2.4.1. If \( A \) is in area 1 and \( B, C, \) and \( D \) are in area 4, let \( \bar{A} = (\hat{x}_1^*(a, b), \hat{x}_2^*(a, b)) \) and \( \bar{D} = (\hat{x}_1^*(a, b) + c - a, \hat{x}_2^*(a, b)) \). Since \( \bar{A}, \bar{B}, \bar{C}, \) and \( \bar{D} \) are all in area 4, \( \tau_k(\bar{C}) + \tau_k(\bar{D}) \geq \tau_k(\bar{A}) + \tau_k(\bar{B}) \). \( \tau_k(\bar{A}) = \tau_k(\bar{A}) \) and \( \tau_k(\bar{D}) \leq \tau_k(\bar{D}) \).

2.2.4.2. If \( A \) and \( C \) are in area 1 and \( B \) and \( D \) are in area 4, then \( \tau_k(\bar{A}) = \tau_k(\bar{C}) \) and \( \tau_k(\bar{B}) \leq \tau_k(\bar{D}) \).

2.2.4.3. If \( A \) and \( D \) are in area 1 and \( B \) and \( C \) are in area 4, let \( \bar{A} = (\hat{x}_1^*(a, b), \hat{x}_2^*(a, b)) \) and \( \bar{D} = (\hat{x}_1^*(a, b) + c - a, \hat{x}_2^*(a, b)) \). Since \( \bar{A}, \bar{B}, \bar{C}, \) and \( \bar{D} \) are all in area 4, \( \tau_k(\bar{C}) + \tau_k(\bar{D}) \geq \tau_k(\bar{A}) + \tau_k(\bar{B}) \). \( \tau_k(\bar{A}) = \tau_k(\bar{A}) \) and \( \tau_k(\bar{D}) \leq \tau_k(\bar{D}) \).

2.2.4.4. If \( A, C, \) and \( D \) are in area 1 and \( B \) is in area 4, then \( \tau_k(\bar{A}) = \tau_k(\bar{C}) \) and \( \tau_k(\bar{B}) \leq \tau_k(\bar{D}) \).

2.2.5. Areas 1 and 3

2.2.5.1. If \( A \) is in area 1 and \( B, C, \) and \( D \) are in area 3, by Lemma 6, \( \phi^*_{k,x_1^*}(\hat{x}_1^*(a, b)) + \phi^*_{k,x_1^*}(c + \delta - a + \hat{x}_1^*(a, b)) \leq \phi^*_{k,x_1^*}(a + \delta) + \phi^*_{k,x_1^*}(c) \). \( \tau_k(\bar{A}) \leq \phi^*_{k,x_1^*}(\bar{A}) \), \( \tau_k(\bar{D}) \leq \phi^*_{k,x_1^*}(\bar{D}) \), and \( \tau_k(\bar{D}) \geq \phi^*_{k,x_1^*}(\bar{D}) \).

2.2.5.2. If \( A \) and \( C \) are in area 1 and \( B \) and \( D \) are in area 3, \( \tau_k(\bar{A}) = \tau_k(\bar{C}) \) and \( \tau_k(\bar{B}) \leq \tau_k(\bar{D}) \).

2.2.5.3. If \( A \) and \( D \) are in area 1 and \( B \) and \( C \) are in area 3, \( \tau_k(\bar{A}) \leq \tau_k(\bar{C}) \) and \( \tau_k(\bar{B}) \leq \tau_k(\bar{D}) \).

2.2.5.4. If \( A, C, \) and \( D \) are in area 1 and \( B \) is in area 3, \( \tau_k(\bar{A}) = \tau_k(\bar{C}) \) and \( \tau_k(\bar{B}) \leq \tau_k(\bar{D}) \).

2.3. All points are in three areas

2.3.1. Areas 1, 2, and 3

2.3.1.1. If \( A \) and \( C \) are in area 1; \( D \) is in area 2; and, \( B \) is in area 3, \( \tau_k(\bar{A}) = \tau_k(\bar{C}) \) and \( \tau_k(\bar{B}) \leq \tau_k(\bar{D}) \).

2.3.2. Areas 1, 3, and 4

2.3.2.1. If \( A \) is in area 1; \( B \) and \( D \) are in area 3; and, \( C \) is in area 4, by Lemmas 6 and 2, \( \phi^*_{k,x_1^*}(\bar{A}) + \phi^*_{k,x_1^*}(\bar{C}) \leq \phi^*_{k,x_1^*}(\bar{A} + \delta) + \phi^*_{k,x_1^*}(\bar{C}) \). Due to the property of lozenge
(i). \( \tau_k(A) - \phi^*_{k,x_2}(a) \leq \tau_k(C) - \phi^*_{k,x_2}(a + \delta) \leq 0 \). \( \tau_k(B) = \phi^*_{k,x_2}(c + \delta), \tau_k(D) = \phi^*_{k,x_2}(c) \).

2.3.2.2. If \( A \) and \( C \) are in area 1; \( B \) is in area 3; and, \( D \) is in area 4, \( \tau_k(A) = \tau_k(C) \) and \( \tau_k(B) \leq \tau_k(D) \).

2.3.2.3. If \( A \) is in area 1; \( B \) is in area 3; and, \( C \) and \( D \) are in area 4, draw line \( \ell \) which is parallel to lines \( AC \) and \( BD \) and passes through point \( \tilde{B} \). Then there are two different cases (see the following figure). Proof is same as 2.2.3.4.

\[ \text{Figure 14. Directional property 1 – Proof 2.3.2.3} \]

2.4. All points are in four areas

2.4.1. If \( A \) is in area 1; \( D \) is in area 2; \( B \) is in area 3; and, \( C \) is in area 4, draw line \( \ell \) which is parallel to lines \( AC \) and \( BD \) and passes through point \( \tilde{B} \). Then there are two different cases.

\[ \text{Figure 15. Directional property 1 – Proof 2.4.1} \]
Figure 15 (a) represents the case when point C is above line ℓ, and figure (b) represents the case when point C is below line ℓ.

Case (a): $\tau_k(A) \leq \tau_k(D)$, $\tau_k(B) = \tau_k(C)$. Hence, $\tau_k(A) + \tau_k(B) \leq \tau_k(C) + \tau_k(D)$.

Case (b): Since all four points $\tilde{A}$, $\tilde{B}$, $\tilde{C}$, and $\tilde{D}$ are in area 4, $\tau_k(A) + \tau_k(B) \leq \tau_k(C) + \tau_k(D)$. Also, $\tau_k(A) = \tau_k(C)$, $\tau_k(B) \geq \tau_k(D)$, and $\tau_k(D) \leq \tau_k(D)$. Therefore, $\tau_k(A) + \tau_k(B) \leq \tau_k(C) + \tau_k(D)$.

So far, we have shown that if $\phi_k(x_k)$ satisfies the directional properties represented in Figure 12, the structure of the optimal policy should look like that in Figure 12, and $v_k(x_k)$ satisfies the directional properties.

By Lemma 8, if $v_{k+1}(x_{k+1})$ satisfies the directional properties in Figure 12, $\phi_k(x_k)$ satisfies the directional properties as well. Therefore, given that $v_{n+1}(x_{n+1})$ satisfies the directional properties (e.g., $v_{n+1}(x_{n+1}) = 0$), by the principle of the mathematical induction, $\phi_k(x_k)$ and $v_k(x_k)$ satisfies the directional properties for $k = n, n - 1, \ldots, 1$. Q.E.D.

The structure of the optimal policy in $(x_{k;1}^i, x_{k;2}^i)$ consists of four areas (see Figure 12). In area 1, only acceptance decisions are made, in area 2, some others are accepted and some others rejected, in area 3, only rejections are made, and finally in area 4, any order in the queue for acceptance is not either accepted or rejected. Due to the submodularity property, the optimal decisions are monotonic in $x_{k;1}^i$ and/or $x_{k;2}^i$. The amount of accepted orders is non-increasing while the amount of rejected orders is non-decreasing in $x_{k;1}^i$ and/or $x_{k;2}^i$.

4.3.2 Directional Property 2: $(g_k^{ij}, x_{k;1}^i)$

In this section, we only consider $g_k^{ij}$ and $x_{k;1}^i$ as variables and assume that all other variables remain fixed. Let $(\tilde{g}_k^{ij}, (g_k^{ij}, x_{k;1}^i), \tilde{x}_{k;1}^i (g_k^{ij}, x_{k;1}^i)) := \arg\max_{(\tilde{g}_k^{ij}, \tilde{x}_{k;1}^i)} \phi_k(\tilde{g}_k^{ij}, \tilde{x}_{k;1}^i)$ s.t. $\tilde{g}_k^{ij} + \tilde{x}_{k;1}^i = g_k^{ij} + x_{k;1}^i$ and $\tilde{\phi}_k(\tilde{g}_k^{ij}, x_{k;1}^i) := \max \phi_k(\tilde{g}_k^{ij}, \tilde{x}_{k;1}^i)$ s.t. $\tilde{g}_k^{ij} + \tilde{x}_{k;1}^i = g_k^{ij} + x_{k;1}^i$.

The following figure represents the structure of the optimal policy (i.e., left figure) and the directional properties in $(g_k^{ij}, x_{k;1}^i)$ two dimensional space. Lozenges (i) and (ii) presented in the figure are same as those in Figure 9.
Proposition 10. If \( \phi_k(x_k) \) satisfies \( \phi_k(A) + \phi_k(B) \geq \phi_k(C) + \phi_k(D) \) in the case of lozenge (i) and \( \phi_k(A) + \phi_k(B) \leq \phi_k(C) + \phi_k(D) \) in the case of lozenge (ii), then 1) the optimal policy in \( (g_{k}^{ij}, x_{k:1}^i) \) plane looks like the structure represented in Figure 15 and 2) \( u_k(z) \) satisfies the condition \( u_k(A) + u_k(B) \geq u_k(C) + u_k(D) \) in the case of lozenge (i) and \( u_k(A) + u_k(B) \leq u_k(C) + u_k(D) \) in the case of lozenge (ii).

Proof of Proposition 10.

Similar to the proof of Proposition 8, first, the concavity implies the existence of the optimal solution, and next, the directional properties imply that \( g_{k}^{ij} + x_{k:1}^i \) is decreasing in \( g_{k}^{ij} + x_{k:1}^i \) and \( x_{k:1}^i \) is increasing in \( g_{k}^{ij} + x_{k:1}^i \). All of these finally result in the structure of the optimal policy shown in Figure 16.

The following is to show that if \( \phi_k(x_k) \) satisfies the directional properties represented in Figure 16, \( u_k(x_k) \) satisfies the directional properties as well.

1. Case of lozenge (i)

1.1. All points are in one area:

1.1.1. If all four points are in area 1, since \( \phi_k(x_k) \) satisfies \( \phi_k(A) + \phi_k(B) \geq \phi_k(C) + \phi_k(D) \) and \( \phi_k(x_k) = \tau_k(x_k), \tau_k(A) + \tau_k(B) \geq \tau_k(C) + \tau_k(D) \).

1.1.2. If all four points are in area 2, \( \tau_k(A) = \tau_k(D) \) and \( \tau_k(B) = \tau_k(C) \).

1.2. All points are in two areas:

1.3. If \( D \) is in area 1 and \( B, C, \) and \( A \) are in area 2, \( \tau_k(B) = \tau_k(C) \) and \( \tau_k(A) \geq \tau_k(D) \).
1.4. If A and C are in area 2 and B and D are in area 1, then let \( \tilde{C} = (\tilde{\tilde{g}}_{ik}^{ij\ast}(C), \tilde{x}_{k;1}^{i\ast}(C)) \); \( \tilde{A} = (\tilde{\tilde{g}}_{ik}^{ij\ast}(C), \tilde{x}_{k;1}^{i\ast}(C) + b - d ) \). Then \( \tau_k(\tilde{A}) + \tau_k(B) \leq \tau_k(\tilde{C}) + \tau_k(D) \). Since \( \tau_k(C) = \tau_k(\tilde{C}) \) and \( \tau_k(A) + \tau_k(C) \geq \tau_k(C) + \tau_k(D) \).

1.5. If A and D are in area 1 and B and C are in area 2, then \( \tau_k(B) = \tau_k(C) \) and \( \tau_k(A) \geq \tau_k(D) \).

1.6. If A, B, and D are in area 1 and C is in area 2, then let \( \tilde{C} = (\tilde{\tilde{g}}_{ik}^{ij\ast}(C), \tilde{x}_{k;1}^{i\ast}(C)) \); \( \tilde{A} = (\tilde{\tilde{g}}_{ik}^{ij\ast}(C), \tilde{x}_{k;1}^{i\ast}(C) + b - d ) \). Then \( \tau_k(\tilde{A}) + \tau_k(B) \leq \tau_k(\tilde{C}) + \tau_k(D) \). Since \( \tau_k(C) = \tau_k(\tilde{C}) \) and \( \tau_k(A) + \tau_k(B) \geq \tau_k(C) + \tau_k(D) \).

2. Case of lozenge (ii)

2.1. All points are in one area

2.1.1. If all four points are in area 1, since \( \phi_k(x_k) \) satisfies \( \phi_k(A) + \phi_k(B) \leq \phi_k(C) + \phi_k(D) \) and \( \phi_k(x_k) = \tau_k(x_k) \), \( \tau_k(A) + \tau_k(B) \leq \tau_k(C) + \tau_k(D) \).

2.1.2. If all four points are in area 2, then \( \tau_k(A) = \tau_k(C) \) and \( \tau_k(B) = \tau_k(D) \).

2.2. All points are in two areas

2.2.1. If B is in area 1 and A, C, and D are in area 2, then \( \tau_k(A) = \tau_k(C) \) and \( \tau_k(B) \leq \tau_k(D) \).

2.2.2. If A and C are in area 2 and B and D are in area 1, then \( \tau_k(A) = \tau_k(C) \) and \( \tau_k(B) \leq \tau_k(D) \).

2.2.3. If A and D are in area 2 and B and C are in area 1, then let \( \tilde{A} = (\tilde{\tilde{g}}_{ik}^{ij\ast}(A), \tilde{x}_{k;1}^{i\ast}(A)) \); \( \tilde{D} = (\tilde{\tilde{g}}_{ik}^{ij\ast}(A) + c - a, \tilde{x}_{k;1}^{i\ast}(A)) \). Then \( \tau_k(\tilde{A}) + \tau_k(B) \leq \tau_k(\tilde{C}) + \tau_k(D) \). Since \( \tau_k(A) = \tau_k(\tilde{A}) \) and \( \tau_k(D) \geq \tau_k(\tilde{D}) \), \( \tau_k(A) + \tau_k(B) \leq \tau_k(C) + \tau_k(D) \).

2.2.4. If B, C, and D are in area 1 and A is in area 2, then let \( \tilde{A} = (\tilde{\tilde{g}}_{ik}^{ij\ast}(A), \tilde{x}_{k;1}^{i\ast}(A)) \); \( \tilde{D} = (\tilde{\tilde{g}}_{ik}^{ij\ast}(A) + c - a, \tilde{x}_{k;1}^{i\ast}(A)) \). Then \( \tau_k(\tilde{A}) + \tau_k(B) \leq \tau_k(\tilde{C}) + \tau_k(D) \). Since \( \tau_k(A) = \tau_k(\tilde{A}) \) and \( \tau_k(D) \geq \tau_k(\tilde{D}) \), \( \tau_k(A) + \tau_k(B) \leq \tau_k(C) + \tau_k(D) \).

So far, we have shown that if \( \phi_k(x_k) \) satisfies the directional properties represented in Figure 16, the structure of the optimal policy should look like that in Figure 16, and \( \nu_k(x_k) \) satisfies the directional properties.

By Lemma 8, if \( \nu_{k+1}(x_{k+1}) \) satisfies the directional properties in Figure 16, \( \phi_k(x_k) \) satisfies the directional properties as well. Therefore, given that \( \nu_{n+1}(x_{n+1}) \) satisfies the directional properties (e.g.,
by the principle of the mathematical induction, $\phi_k(\bar{x}_k)$ and $v_k(\bar{x}_k)$ satisfies the directional properties for $k = n, n-1, \ldots, 1$. Q.E.D.

The structure of the optimal policy in $(g_{k}^{ij}, x_{k:1}^i)$ consists of two areas (see Figure 16). In area 1, no order is assigned to the $j$-type server, and in area 2, some orders are assigned. The optimal decisions are also monotonic in $g_{k}^{ij}$ and/or $x_{k:1}^i$. The amount of assigned orders is non-increasing in $g_{k}^{ij}$ and/or $x_{k:1}^i$.

4.3.3 Directional Property 3: $(g_{k}^{ij}, x_{k:2}^i)$

Unlike $(x_{k:1}^i, x_{k:2}^j)$ and $(g_{k}^{ij}, x_{k:1}^i)$ cases, $(g_{k}^{ij}, x_{k:2}^i)$ does not have a consistent directional property because it does not satisfy Proposition 8. Therefore, the optimal policy at a given $k$ should form one among two possible structures like the following figures depending on the shape of

$$\left(\xi_{k:1}^i, x_{k:2}^i, g_{k}^{ij}(\xi_{k:2}^i)\right) = \arg\max (\beta_{k}^{ij}(\xi_{k:2}^i) \phi_k(\beta_{k}^{ij}, \xi_{k:2}^i)) \text{ s.t. } \tilde{g}_{k}^{ij} + \tilde{x}_{k}^{i} = g_{k}^{ij} + x_{k:2}^i.$$ 

![Figure 17. Structure of the optimal policy in $(g_{k}^{ij}, x_{k:2}^i)$](image)

The structure of the optimal policy in $(g_{k}^{ij}, x_{k:2}^i)$ is similar to that in $(x_{k:1}^i, x_{k:2}^j)$ except the shape of area 1 and 4. The monotonicity of the optimal decisions is also preserved like $(x_{k:1}^i, x_{k:2}^j)$ case.

4.4 Structure of the optimal policy

In previous sections, we already showed the structure of the optimal policy in some dimensions - $(x_{k:1}^i, x_{k:2}^j)$, $(g_{k}^{ij}, x_{k:1}^i)$, and $(g_{k}^{ij}, x_{k:2}^i)$. In this section, we show the structure of the optimal policy in remaining two-dimensional spaces which include $(x_{k:1}^i, x_{k:1}^i)$, $(x_{k:2}^j, x_{k:2}^j)$, $(x_{k:1}^i, x_{k:2}^i)$, $(x_{k:1}^i, g_{k}^{ij})$,
(x_{k:2}^{ij}, g_k^{ij}), (g_k^{ij}, g_k^{ij}), (g_k^{ij}, g_k^{ij}), and (g_k^{ij}, g_k^{lm}). Also, the overall structure of the optimal policy in the original state space, \( \mathbb{R}^{2i+1} \), is discussed.

First, the structure of the optimal policy in \((x_{k:1}^{i}, x_{k:1}^{i}), (x_{k:2}^{i}, x_{k:2}^{i}), (x_{k:1}^{i}, x_{k:2}^{i})\) are shown in Figure 18 (a). In areas 1 and 3, only one variable is changed (i.e., rejecting, accepting, or assigning). In area 2, both variables may be moved together, while in area 4, no variable is changed (i.e., stay put). Figure 18 (b) represent the structure of the optimal policy in \((g_k^{ij}, g_k^{ij})\) and \((g_k^{ij}, g_k^{lm})\). In areas 1 and 3, one variable is to be changed (i.e., assigning), in area 2, stay put is the optimal decision, and finally, in areas 4 and 5, both variables may be moved together.

![Figure 18. Structure of the optimal policy in \((x_{k:1}^{i}, x_{k:1}^{i}), (x_{k:2}^{i}, x_{k:2}^{i}), (x_{k:1}^{i}, x_{k:2}^{i}), (g_k^{ij}, g_k^{ij})\) and \((g_k^{ij}, g_k^{lm})\)](image)

Figure 19 (a) represent the cases of \((x_{k:1}^{i}, g_k^{ij})\) and \((x_{k:2}^{i}, g_k^{ij})\). In area 1, stay put is the best decision, in areas 2 and 4, only one variable is associated with changing values, and lastly in area 3, both variables are moved. Figure 19 (b) shows the decisions related to selecting right servers to assign for given orders. In areas 1 and 3, only one type should be selected, while in area 4, orders may be split into two types of server.
Figure 19. Structure of the optimal policy in $(x^i_{k:1}, g^i_k)$, $(x^i_{k:2}, g^i_k)$, and $(g^i_k, g^i_k)$.

Unlike the cases of $(x^i_{k:1}, x^i_{k:2})$, $(g^i_k, x^i_{k:1})$, and $(g^i_k, x^i_{k:2})$ of which the structure of the optimal policy we already showed in the previous section, decisions in these two-dimensional planes (i.e., changing the values of the focal variables of the plane) sometimes change the values of other variables which are not in the focal plane and so assumed to be fixed. For example, consider the case of $(x^i_{k:1}, x^i_{k:1})$. The number of orders in the queue for assignment, $x^i_{k:1}$, cannot be changed without affecting the number of working servers, $g^i_k$, which is not defined in the $(x^i_{k:1}, x^i_{k:1})$ plane and is assumed to be fixed. Such changes in other variables’ values may alter the structure of the optimal policy – they may move the lines and intersections (i.e. thresholds) of the optimal policy – while the structural properties remain intact. Those changes in the optimal policy also can be characterized. In general, due to the submodularity property, increasing the values of other variables is associated with decreasing (or non-increasing) the optimal values of focal variables while decreasing the values of other variables is associated with increasing (or non-decreasing) the optimal values of focal variables.

It is not easy to visualize the structure of the optimal policy due to the multidimensionality of the state space. However, based on the properties of the objective function and value function that we find, we can show the general structure of the optimal policy as a whole in the original state space, $\mathbb{R}^{2I+IJ}$ (see the following figure).
Figure 20. Structure of the optimal policy in the original state space $\mathbb{R}_+^{2I+IJ}$

There is a subset of $\mathbb{R}_+^{2I+IJ}$, area 1, which always can achieve the global optimum decision at time $k$. This area is characterized by sufficient orders and low server utilization. The second subset (i.e. area 2) is when there are a few available servers (i.e., high utilization) but too many orders in the queues. In this case, in order to decrease the waiting penalties, some orders may be rejected or even may be assigned to the non-best matching servers. Area 3 is characterized by the high level of server utilization and the low level of orders in the queues, which enable decision makers to postpone decisions for acceptance and assignment while keeping current orders to wait and not worrying about the risk of low utilization too much. Lastly, in area 4, there are many idle servers so that some orders need to be assigned to the non-best matching servers to hedge the risk of low utilization of servers. Lines of the structure are drawn being consistent with the submodularity properties.

5 Approximate Policies and Computations

Although Theorems 2 (Concavity) and 3 (Submodularity) and Propositions 9 and 10 (Directional Properties) give us the structural properties of the optimal policy, they do not tell us the exact policy – e.g., what the threshold is to accept type 1 order or to assign type 2 order to type 3 server, etc. These must be obtained by solving the dynamic equations through standard techniques such as value iteration. However, our problem is difficult to solve even for the moderate-size problems due to its complexity. Therefore, in this chapter, we focus to develop good approximate policies.

Our basic idea for approximate policies is to use the preemptive decisions. More specifically, we approximate the original profit-to-go function, $\phi_k^2(x_k) = E[v_{k+1}(x_{k+1}, x_{k-1}, ..., x_{k-1}, x_{k-2}, w_k^1 + Z_{k+1}, x_{k-2}^2 w_k^2 + Z_{k+1}^2, x_{k-2}^1 w_k^1 + Z_{k+1}^1, g_{k-1}^{11} l_{k-1}, g_{k-1}^{12} l_{k-1}^1, ..., g_{k-1}^{l1} l_{k-1}^l)]$, using the value function of the preemptive problem where reassignment decisions are possible. Since the preemptive problem is much
less complex than the original non-preemptive problem, this way allows us to tackle much larger problems and to reduce the computation time.

### 5.1 Preemptive Problem

Preemptive assumption allows the decision makers not to worry about wrong assignment decisions and to focus on myopic assignments, since at every time (i.e., decision epoch), orders including ones already assigned to servers can be reassigned. Therefore, in this case, there is no reason to intentionally leave any server idle – servers should be assigned as many as possible.

Since orders are reassigned at every period, we can model that at the end of period unfinished orders are back to the queue waiting for assignment. Thus we have

\[ x_{k+1;1} = x_{k;1} + A_k - \sum_{j=1}^{L} U_k^{ij} (1 - L_k^j), \quad k = 1, 2, \ldots, n; \quad i = 1, 2, \ldots, l \]

Since assignments only last during one period, the state variable, \( G_k^{ij} \), is always 0. Therefore, a state can be denoted only by \( (x_{k;1}, x_{k;2}) \) without \( g_k \). Although assignment variables do not represent the problem state anymore, assignment decisions at time \( k \), \( U_k^{ij} = Q_k^{ij} - G_k^{ij} = Q_k^{ij} \), in this case, still affect the expected profit. Let \( \tilde{x}_k = (x_{k;1}, x_{k;2}) \) and \( \tilde{y}_k = (y_{k;1}, y_{k;2}) \). And we use \( Q_k^{ij} \) instead of \( U_k^{ij} \) for being consistent with the original model. Then, the expected profit for the remaining periods by making preemptive decisions at time \( k \) is:

\[
\hat{\psi}_k \left( \tilde{x}_k, \tilde{y}_k, q_k \right) = -\sum_{i=1}^{l} r_i^k x_{k;1}^i + \sum_{i=1}^{l} \left( (r_i^k - b_{k;1}^i) y_{k;1}^i - b_{k;2}^i \mathbb{E}[y_{k;2}^i W_k^1] \right) \\
+ \sum_{i=1}^{l} \sum_{j=1}^{l} (r_i^k - c_{ij}^k) q_{ij}^k \\
+ \mathbb{E} \left[ \tilde{q}_{k+1}^{1} (y_{k;1}^1 + \sum_{j=1}^{l} q_k^{1j} y_{k;1}^j, y_{k;2}^1 + \sum_{j=1}^{l} q_k^{2j} L_k^j, \ldots, y_{k;1}^1 + \sum_{j=1}^{l} q_k^{lj} y_{k;2}^j, y_{k;2}^1 W_k^1 \\
+ Z_{k+1}^{1}, y_{k;2}^2 W_k^2 + Z_{k+1}^{2}, \ldots, y_{k;2}^l W_k^l + Z_{k+1}^{l}) \right] 
\]

Let \( \hat{\psi}_k \left( \tilde{x}_k, \tilde{y}_k, q_k \right) = \max \hat{\psi}_k \left( \tilde{x}_k, \tilde{y}_k, q_k \right) \)

s.t.

\( \hat{q}_{k;1}^{i} + \hat{q}_{k;2}^{i} + \sum_{j=1}^{l} q_{ij}^k \leq \tilde{x}_{k;1}^i + \tilde{x}_{k;2}^i, \quad i = 1, 2, \ldots, l \)

\( \hat{q}_{k;2}^i \leq \tilde{x}_{k;2}^i, \quad i = 1, 2, \ldots, l \)
Approximate Policy using the Preemptive Value Function (AP)

Approximate policy using the Preemptive value function (AP) use the value function of the preemptive problem, \( \tilde{\psi}_{k+1}(\tilde{y}_k) \), instead of the original value function, \( v_{k+1}(y_k) \). This policy can be conceptualized as making current decisions assuming that preemptive decisions can be made from the next period. Let \( \tilde{\phi}_k(y_k) \) be the objective function to find AP:

\[
\tilde{\phi}_k(y_k) = \sum_{i=1}^{l} (r^i_k - b^i_{k:1})y^i_{k:1} - b^i_{k:2}E[y^i_{k:2}W^i_k] + \sum_{i=1}^{l} \sum_{j=1}^{j} (r^i_k - c^i_k)q^i_k
\]

Then the maximization problem to find \( y_k(x_k)^* \) can be formally represented as

\[
y_k(x_k)^* = \arg\max \tilde{\phi}_k(y_k) \text{ s.t. (cs1)} - (cs6).
\]

A more costly variant of this approximation is to use the preemptive value function when calculating the profit-to-go function at time \( k + 2 \) instead of at time \( k + 1 \). We call this variant as Approximate Policy using the Preemptive value function after 2 periods (AP2), which should perform better than or at least equal to AP.

5.3 Computational Experiments

In order to assess the effectiveness of our approximate policies, we implement and apply them to a variety of test problems. We measure the effectiveness of the proposed policies in terms of the profit that it generates for each problem instance relative to the profit for the exact policy. Since the sample paths are always changed by the decisions being made at each period, the exactly same problem instances cannot be used to compare the performances of two different policies. Therefore, we use the average profits of policies over 1 million problem instances for the same problem scenario – having the same set of parameter values – for performance comparison.

Since problem difficulty and policy performance depend on various key problem parameters, we test the performance of our heuristics over many different problem scenarios, defined by varying following key problem parameters.
• Number of service types and server types. Problems with more types are harder to solve due to increased dimensions and complexity. For our experiment, we consider problems with (2 – 3) and (3 – 3) service and server type pairs.

• Maximal queue length and server capacity: the amount of orders in a waiting queue can be limited. Problems with larger queue lengths are more difficult to solve. Similarly, larger server capacity is also associated with more complex problems. For our computations, we consider problems with (2 – 2), (4 – 1), and (10 – 1) max queue length and server capacity pairs.

• Relative revenue of service types: When some service types generate much more revenue than others, a wrong acceptance decision may significantly decrease the profit of approximate policies. In order to investigate this effect, we define the ratio of the minimal revenue to the maximal revenue of service type and consider two values of revenue ratios – small (0.25) and large (0.75).

• Relative service cost of service and server pair: Although all servers are cross-trained so that they can provide any type of service, disparity in their performance exists – service costs are different in our case. We assume that between service types and server types, there are well-matching pairs of which performance is better than other pairs, non-matching pairs. As the performance gaps between the well-matching pair and the non-matching pair become larger, a wrong assignment decision more significantly decreases the profit of approximate policies. However, at the same time, if the gap becomes too huge, assignment decisions may become trivial. In order to study this effect, we consider two values of service cost ratios (i.e., min service cost / max service cost) – small (0.25) and large (0.75).

• Relative service time of service and server pair: Not only service costs are different, but also service times may depend on the capability of servers. We assume that a more capable server complete a job faster than less capable one. In our experiments, the probability for service completion is defined using the following function: \(1/(uniform\_factor^{*}n + 1)\), for \(i = 1, 2, \ldots, n\); or, \(((uniform\_factor - 1)^{*}n + 1)/(uniform\_factor^{*}n + 1)\), for \(i = 0\), where \(i\) is the capacity of available servers, which complete current jobs at the end of period, \(n\) is the total capacity of working servers at the beginning of period, and \(uniform\_factor\) is a parameter of which value is between 1 to infinity. When \(uniform\_factor\) is 1, service completion distribution is a uniform distribution, while as \(uniform\_factor\) gets larger, the distribution becomes more biased so that service completion time gets longer. In order to investigate the effect of the disparity in service completion time, we consider three cases of the minimal \(uniform\_factor\) vs. the maximal \(uniform\_factor\) – 1:1, 1:5, and 1:20.
• Relative waiting penalty of service types: The disparity in waiting penalties of service types also affects acceptance/assignment decisions. However, too huge gap between service types may make also problems too trivial. In order to study this effect, we define the ratio of waiting penalty (min waiting penalty / max waiting penalty) and consider two values – small (0.25) and high (0.75).

• Arrival rates: We consider two arrival rates – low (0.25) and high (1). We do not differentiate this value across service types since our pilot experiments did not show significant difference.

• Abandonment rates: We use a uniform distribution for abandonment and do not vary this by service type since our pilot experiments did not show significant difference.

Our study considers total 192 problem scenarios – combinations of some parameter values. For each scenario, 1 million problem instances are generated and tested for each policy – the exact policy, AP, and AP2. The effectiveness of our approximate policies are shown by comparing the average performance of each policy over 1 million instance with the average performance of the exact policy for the same problem scenario. We implemented problem generators and policies in C programming language. Our programs were run on a Unix machine of which CPU is 3.2 GHz and RAM is 4 Gbyte. The following table shows the experiment results.

Table 3. Performance of AP and AP2

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25 0.25 0.25 0.25</td>
<td>2312.2 2300.7 2311.3</td>
<td>0.49 0.04</td>
<td>2300.9 2298.3 2300.1</td>
<td>0.12 0.04</td>
<td>2300.3 2299.6 2299.6</td>
<td>0.03 0.03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.25 0.25 0.25 0.25</td>
<td>8389.4 8385.3 8388.1</td>
<td>0.05 0.02</td>
<td>7851.1 7763.4 7832.7</td>
<td>1.13 0.24</td>
<td>7610.2 7510.1 7600.8</td>
<td>1.33 0.12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.25 0.25 0.75 0.25</td>
<td>2278.7 2277.8 2278.0</td>
<td>0.04 0.03</td>
<td>2194.0 2104.8 2175.9</td>
<td>4.24 0.83</td>
<td>2179.2 2039.3 2167.9</td>
<td>6.86 0.52</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.25 0.25 0.75 0.25</td>
<td>7859.3 7858.4 7859.3</td>
<td>0.01 0.00</td>
<td>6727.3 6686.5 6711.6</td>
<td>0.61 0.23</td>
<td>6323.5 5996.5 6199.3</td>
<td>3.95 0.55</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.25 0.75 0.25 0.75</td>
<td>976.4 967.3 976.4</td>
<td>0.94 0.00</td>
<td>965.3 964.1 964.7</td>
<td>0.13 0.07</td>
<td>965.2 965.2 966.2</td>
<td>0.00 0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.25 0.75 0.25 0.75</td>
<td>3059.5 3054.3 3057.8</td>
<td>0.17 0.06</td>
<td>2680.9 2567.5 2677.4</td>
<td>4.41 0.13</td>
<td>2655.5 2653.8 2654.4</td>
<td>0.06 0.04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.25 0.75 0.75 0.75</td>
<td>943.5 942.8 943.2</td>
<td>0.07 0.03</td>
<td>858.9 770.9 840.3</td>
<td>11.41 2.21</td>
<td>852.4 707.7 839.4</td>
<td>20.45 1.55</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.25 0.75 0.75 0.75</td>
<td>2582.2 2575.7 2579.1</td>
<td>0.26 0.12</td>
<td>2055.8 1903.2 2053.6</td>
<td>8.02 0.10</td>
<td>1993.7 1931.8 1992.5</td>
<td>3.20 0.06</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.75 0.25 0.25 0.25</td>
<td>2126.4 2125.2 2126.1</td>
<td>0.06 0.02</td>
<td>2092.9 2087.9 2091.8</td>
<td>0.24 0.05</td>
<td>2093.9 2092.8 2093.8</td>
<td>0.05 0.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.75 0.25 0.25 0.25</td>
<td>8008.5 8001.7 8006.0</td>
<td>0.09 0.03</td>
<td>7452.1 7399.6 7441.6</td>
<td>0.71 0.14</td>
<td>2084.7 2079.0 2083.6</td>
<td>0.28 0.06</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.75 0.25 0.75 0.25</td>
<td>2102.0 2100.7 2101.6</td>
<td>0.06 0.02</td>
<td>2002.5 1945.9 1988.8</td>
<td>2.91 0.69</td>
<td>1983.1 1884.4 1974.1</td>
<td>5.24 0.46</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.75 0.25 0.75 0.25</td>
<td>7540.7 7536.1 7539.3</td>
<td>0.06 0.02</td>
<td>6398.9 6369.6 6390.8</td>
<td>0.19 0.13</td>
<td>5823.7 5693.9 5787.7</td>
<td>2.28 0.62</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.75 0.75 0.25 0.25</td>
<td>791.7 790.9 791.0</td>
<td>0.23 0.09</td>
<td>757.9 755.3 757.8</td>
<td>0.34 0.00</td>
<td>758.1 756.9 758.0</td>
<td>0.16 0.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.75 0.75 0.25 0.25</td>
<td>2673.3 2667.2 2671.4</td>
<td>0.09 0.07</td>
<td>2262.4 2174.2 2257.8</td>
<td>4.06 0.20</td>
<td>2235.7 2231.7 2235.0</td>
<td>0.18 0.03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.75 0.75 0.75 0.25</td>
<td>767.1 766.4 766.9</td>
<td>0.09 0.03</td>
<td>665.8 605.4 654.1</td>
<td>9.98 1.78</td>
<td>657.5 551.6 644.3</td>
<td>19.20 2.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.75 0.75 0.75 0.75</td>
<td>2251.0 2250.6 2250.6</td>
<td>0.02 0.02</td>
<td>1664.3 1609.0 1658.9</td>
<td>3.43 0.32</td>
<td>1600.4 1577.4 1600.5</td>
<td>1.46 0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- # of services types = 2, # of server types = 3, Max queue = 10, Server Capacity = 1
- # of services types = 2, # of server types = 3, Max queue = 4, Server Capacity = 1
- # of services types = 2, # of server types = 3, Max queue = 1, Server Capacity = 1
- # of services types = 2, # of server types = 3, Max queue = 20, Server Capacity = 1
Lastly, the maximal queue length is found to increase the gap performance, which implies that as the drawbacks of myopic decisions policies may suffer from low performance. All findings are somewhat intuitive since they are all related to the drawbacks of myopic decisions – preemptive decisions are myopic in a non-preemptive situation. However, interestingly, the marginal impact seems to diminish as the queue length increases.

Experimental results show that our AP and AP2 relatively perform well on average. The average gaps for AP and AP2 over total problem instances are 1.9% and 0.2%, respectively.

On average, performance gaps get larger as disparities in service time, revenue, and waiting penalty get larger, which implies that our approximate policies may not work well when huge disparities exist in those parameters, though they perform reasonably well in our extensive set of problem scenarios. Also, the experimental results show that when the arrival rates of orders are very low our approximate policies may suffer from low performance. All findings are somewhat intuitive since they are all related to the drawbacks of myopic decisions – preemptive decisions are myopic in a non-preemptive situation. Lastly, the maximal queue length is found to increase the gap performance, which implies that as the number of alternative decisions get increased, the problem becomes more complex and may negatively affect the performance of approximate policies. However, interestingly, the marginal impact seems to diminish as the queue length increases.
The following figure shows the maximal gap performance (i.e., worst case performance) of approximate policies by key problem parameters. The results are more or less same as those of the average gap. One noticeable thing is that AP is somewhat sensitive to problem parameters while the performance of AP2 is quite stable – AP shows sometimes relatively large gap in some scenarios (max gap = 20.4 \%) and correspondingly some level of performance variations (stddev = 3.87\%), while AP2 performs well in almost all scenarios and show relatively low variations (stddev = 0.41\%; max gap = 2.2 \%).

**Figure 21. Average performance gap by key problem parameter**
In summary, our approximate policies using preemptive value function seem to perform well in situations where disparities among the above problem key parameters are relatively low or moderate.

6 Conclusion

In this paper, we study the dynamic order acceptance and resource assignment problem of a services system which provides heterogeneous services with heterogeneous servers. We formulate the problem into a dynamic and stochastic programming model, define the structural properties of the optimal policy by analyzing the concavity, submodularity, and directional properties, propose approximate policies, and conduct extensive computational experiments to show the effectiveness of the proposed approximate policies.

We contribute to the literature by incorporating the concept of postponable acceptance decisions in the acceptance and assignment model. In our problem, the decision maker can accept or reject orders more strategically and can increase his/her profits by postponing order acceptance decisions and waiting for more profitable orders. However, at the same time, this postponement may result in lost orders and low utilization of the system and may harm the profits. We model this tradeoff between the benefit and
risk of postponing decisions. We are motivated by a practical problem, multi-project selection and assignment problem in the software service industry. However, the model we develop here can be used in similar practical problems where decisions can be postponed for improving expected gains while those postponements are also associated with risks. Related practical problems may include but not be limited to accepting patients, recruiting students or employees, and accepting make-to-orders requests.

Our study is not without limitations. First limitation lies in our proposed approximate policies. Even though our approximate policies are much less expensive to calculate than the exact optimal policy, they also have the scalability problem – the computation time also dramatically increases as the problem size increases. Therefore, our proposed policies may not be appropriate for very large sized problems. Second limitation is that we use the continuous state space although most practical problems including the one from what we are motivated have a discrete state space. It is known that continuous space approximation may produce a wrong result (Ahn et al. 2005). However, discrete state spaces usually make the problem intractable to analyze, and unfortunately our case is this case due to its high dimensions of our state space. Therefore, although it may have problem, the continuous space approximation is necessary in this study.
References


