Eindhoven University of Technology

MASTER

Resource sharing facilitated via central cloud in multi-tenant environment

Kalwar, A.

Award date:
2021

Link to publication

Disclaimer
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

Technische Universiteit Eindhoven
University of Technology

Department of Industrial Engineering and Innovation Sciences
MSc. Manufacturing Systems Engineering
Operations Management and Logistics

Resource sharing facilitated via central cloud in multi-tenant environment

Master Thesis

Akash Kalwar
Student ID: 1349333

Thesis supervisor:	Dr. Yingqian Zhang, IS, TU/e
	Dr. A.E.Akcay, OPAC, TU/e
	K. Herps, Brainport Industry campus
Assessor:	Dr. Q.V. Dang, OPAC, TU/e

Eindhoven, March 2021

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Declaration concerning the TU/e Code of Scientific Conduct for the Master's/PDEng/PhD Thesis

I have read the TU/e Code of Scientific Conduct[i].

I hereby declare that my Master's /PDEng/PhD-thesis has been carried out in accordance with the rules of the TU/e Code of Scientific Conduct.

Date:	24/03/2021
Name:	A. Kalwar
ID-Number:	1349333

Signature:

Kalwar

Submit the signed declaration to CSA IE&IS Atlas 3.331 or csa.ieis@tue.nl.

[i] See: https://www.tue.nl/en/our-university/about-the-university/integrity/scientific-integrity/
The Netherlands Code of Conduct for Academic Practice of the VSNU can also be found here.
More information about scientific integrity is published on the websites of TU/e and VSNU.

Acknowledgement

I would like to thank few people for providing their valuable feedback to shape this underlying research. I would like to thank Dr. Yingqian Zhang and Nitish Singh for allowing me to work on this interesting project. Being my supervisor, yingqian always had time to brainstorm ideas and guide me in the right direction. Special thanks to Nitish for consistently questioning me with different angles during weekly meetings which was a great learning experience and helped me fetch out various managerial insights. This project address the challenges of sharing resources among multiple companies inhabited in the close vicinity.

I would also like to thank Alp Akcay and Koen herps for their guidance and feedback during routine meetings. Furthermore, I would like to thank J. Chen for always having time to listen, brainstorm on ideas. Also, few of my friends with whom I had multiple informal discussions on the progress which helped me think through different perspectives. Finally, I would like to thank my family for their financial and emotional support throughout the journey. Their constant faith kept me motivated to accomplish my dream.

Abstract

Brainport Industries Campus in the Netherlands is a joint venture by high-tech suppliers to work under the same roof for carrying out production. Increasing individualized customer requirements and decreasing production cycle time makes it hard for tenants to efficiently plan resources. By working in close vicinity tenants can help each other by sharing common resources, together invest in high notch resources by promoting collaboration to tackle operational and market uncertainties. BIC has infrastructure of $Industry$ 4.0 with capabilities to fetch real time status of resources. Such an infrastructure can be used to automate the concept of resource sharing with the help of central cloud without any human interference.

In this paper, we develop mechanism to share resources. A conceptual idea is proposed for sharing which is further classified into framework. For scheduling, we develop optimization model that makes scheduling decision based on cost and trust factor. We show that our proposed model helps requester tenant to get resource at cheapest cost from trustworthy provider. Proposed model is flexible to handle multiple time windows and can be extended with more requirements. We also develop trust value update mechanism where trust value of each company is updated at each planning horizon.

Table of Notations

C	$\Longrightarrow$	Set of companies participating in sharing
P	$\Longrightarrow$	Set of companies providing resources, i.e. providers
R	$\Longrightarrow$	Set of companies requesting for resources, i.e. requesters
i	$\Longrightarrow$	Index representing task
N	$\Longrightarrow$	Set of overall tasks
R_i	$\Longrightarrow$	Represent requester who has initiated the task i
j	$\Longrightarrow$	Index representing subtask in task
J	$\Longrightarrow$	Set of overall subtasks
Z_i	$\Longrightarrow$	Set representing subtasks in each task i
b_i	$\Longrightarrow$	Representing number of subtasks in each task
k	$\Longrightarrow$	Index representing unique machine id at shop floor
M	$\Longrightarrow$	Set of machines at shop floor
M_R	$\Longrightarrow$	Set of machines at requester R
M_P	$\Longrightarrow$	Set of machines at provider P
A_k	$\Longrightarrow$	Set representing capabilities of machine k
a_k	$\Longrightarrow$	Total availability of machine in hours
H_n	$\Longrightarrow$	Represents planning horizon
e	$\Longrightarrow$	Index representing idle time window
e_k	$\Longrightarrow$	Set representing number of idle time windows on each machine k
$w^e_{k_l}$	$\Longrightarrow$	Lower bound of available time window e on machine k
$w^e_{k_u}$	$\Longrightarrow$	Upper bound of available time window e on machine k
$t_{i,j}$	$\Longrightarrow$	Processing time of subtask j in task i
r_k	$\Longrightarrow$	Hourly price of machine k
f_C	$\Longrightarrow$	Trust value of company C
$rate^{R,P}_{i,j}$	$\Longrightarrow$	Rating provided by requester R to provider P for accomplishing subtask j of task i
n_P	$\Longrightarrow$	Total number of subtasks allocated to each provider P until horizon H_{n-1}
m_P	$\Longrightarrow$	Number of subtasks j allocated to provider P in horizon H_n

Contents

1 Introduction **1**

2 Literature Review **3**
2.1 Synopsis 3
2.2 Motivation 6

3 Problem Description **7**
3.1 Problem context 7
3.2 Problem Definition 7
3.3 Problem Characteristics 8
3.4 Conceptual Framework 9

4 Mathematical Formulation **11**
4.1 Optimization Model 11
4.2 Trust value update mechanism 16

5 Experiment **18**
5.1 Computation time 19
5.2 Analysis of weight factor α 20
5.3 Sensitivity Analysis 25

6 Extensions and Conclusion **27**
6.1 Suggestion and extension 27
6.2 Future research 27
6.2.1 Full sharing 27
6.2.2 Trust Mechanism 27
6.3 Conclusion 28

Appendix **30**

A **31**
A.1 Travelling time Matrix 31
A.2 Machine Capabilities 32

List of Figures

3.1 Framework . . . 9

4.1 Scheduling allied to cost factor . . . 15
4.2 Scheduling allied to trust factor . . . 15
4.3 Scheduling allied to equal weightage of trust and cost factor . . . 15
4.4 Update mechanism . . . 16

5.1 Cost paid by requesters in each instance . . . 19
5.2 Time to obtain 1st feasible solution in each instance . . . 19
5.3 Total cost paid by requesters after 7 days of sharing . . . 21
5.4 Average machine utilization rate . . . 22
5.5 Overall profit gained by providers . . . 22
5.6 Scheduling results for day 2 with equal weightage of cost and trust factor i.e. $\alpha = 0.5$ . . . 23
5.7 Average machine utilization . . . 25
5.8 Profit gained by each company per machine . . . 25

A.1 Travel time . . . 31

List of Tables

4.1 Requesters input data 14
4.2 Providers input data 14
4.3 Trust value update 17

5.1 Input data 18
5.2 Computational time 20
5.3 Results showing the effect of setting the weight parameter, $\alpha = 1, 0, 0.5$ in optimization model 24
5.4 Updated trust values after each planning horizon 24
5.5 Results of companies being homogeneous, moderately heterogeneous and highly heterogeneous with respect of machine 26

A.1 Capabilities of machines 32

Chapter 1

Introduction

In today's technologically complex world, emerging competitors, customers fluctuating demands and on time delivery of requested orders in a short period of time pose manufacturers to face numerous planning challenges. Especially in high-mix, low-volume market with long production lead time, where capacity and workforce planning is done in view of forecast [Sarker, 2002]. Due to competition between Industries on the global platform, multiple companies are willing to provide better quality products or services at the least possible cost which results into unpredictable customer order behaviour pattern. This has a negative effect on company's short term, mid-term and long term production as well as capacity planning. Since, long term planning has to be done foreseeing the future demands that includes investment in infrastructure, resources required to manufacture, workforce etc. If capacity is greater demand for a planned horizon, it may result into resource overcapacity(excess available resources than required to produce customer orders), violating the principle of resource efficiency. In a similar fashion, if capacity is less than demand for a planned horizon, it may result into resource undercapacity(less resources than required to produce customer orders) which leads to lost orders as well as reduce customer service level. Customers ever increasing volatile individualized product demand will never stop, infact high tech companies have to figure out better business practices to tackle uncertainties of dynamic market.

Brainport Industries Campus (BIC) is a new high technology campus constructed in Eindhoven, Netherlands. BIC is known as Europe's leading innovative top technology region and is a subset of fourth industrial revolution that encompasses area classified as smart city. Industry 4.0 is the trend towards automation and data exchange in manufacturing technologies. Large scale machine-to-machine communication and internet of things are integrated for increased automation. Such an infrastructure facilitates sharing of real time information. Information transparency provide operators with vast amount of useful information to make appropriate decision. Campus is comprised of original equipment manufacturers(OEMs), small and medium size enterprises(SME's), suppliers and knowledge institutes, providing high-tech companies an opportunity to work under the same roof cooperatively and collaboratively, granting an access to each other's physical and fiscal facilities to develop an efficient working environment. BIC facilitates such an environment where uncertain market fluctuations can be easily damped using each others strength. To be more precise, strength is referred to "Sharing Resources", where companies with temporary over-capacity can offer their unused resources to companies with temporary under-capacity during market discrepancy. However, it is known that sharing of resources can be beneficial, but arises a question 'how'?

Cloud manufacturing (CMfg) has gathered more and more attention from researchers and industries in the past few years, as it can visualize various manufacturing resources on the shop floor and build a large shared pool of resources to deliver on demand manufacturing services to users [Chen et al., 2019]. Further, we refer companies who need resources as requester and companies who can provide resources as provider. In general cloud has centralized and decentralized operating mode. In centralized mode, cloud verify requester's requirements and select the services to satisfy their request. Requester do not need to select the provider on it's own, instead cloud select the best match. Whereas in decentralized mode, requester has an autonomy to select available resources on cloud with transparent information provided from provider. Both operating mode are popular in manufacturing industries and e-commerce business platforms. Compared to decentralized way of working, centralized mode has its own perks, due to its ability of having global view on different available resources. Although in decentralized mode requester can select its self-satisfied resources for its job but it is not globally optimal [Liu et al., 2019]. Especially for manufacturing resource like CNC machine, where job has to undergo multiple machining process. For a complex job with multiple sequential dependent machining processes, it can be painful for requesters to individually select multiple capable machines available with distinct providers at dissimilar time. Therefore to take an advantage of such platform where requesters, providers are located in close vicinity and global efficient solution can be generated through collective information. We consider centralized operating mode to develop resource sharing model using concept of CMfg.

In a multi-tenant setting, it is crucial to develop a back-end that meets desired requirements of requesters and also utilize providers resources efficiently. The main aim of this research is to show
(1) How can resources be shared amongst tenants ?
(2) How to conduct scheduling of tasks to resources ?
(3) How profitable resource sharing can be for tenants ?

This research is done towards partial fulfillment of the requirements for Manufacturing Systems Engineering Master track specialized in Operations Management and Logistics at Eindhoven University of Technology that has been carried out for BIC. Main contributions of this paper are summerized below:
(1) Optimization model has been developed for scheduling and sequencing of requests with varied capability requirements on heterogeneous CNC Machines.
(2) Trust value update mechanism has been developed.
(3) Framework has been developed to manifest resource sharing.

Further, this paper is organized as follows. In chapter 2, literature survey of past work has been carried out. In chapter 3, we define the problem, formalize the notation and explain developed framework to facilitate sharing. In chapter 4, we develop optimization model and trust update mechanism. In chapter 5, we carry out various experiments on the model by creating different scenarios and further analyze the output results. In chapter 6, we present our conclusion and future research directions.

Chapter 2

Literature Review

2.1 Synopsis

To overcome the challenges caused by market fluctuation, additional effort needs to posit to balance the logistics and production capacities. [Becker and Stern, 2016] proclaims that a trade-off has to be made between production and logistics performance on the one hand and economic efficiency on the other hand. Most of the companies invest on additional capacities to overcome uncertain demand peaks. Many make managerial decision and endure lost sales considering negative economic impact of installation and operation costs due to seasonal demand patterns. Some companies invest in building inventories foreseeing huge upturn in the demand pattern to cope with fluctuation.[Becker and Stern, 2016, Benjaafar et al., 2019, Freitag et al., 2015, Renna and Argoneto, 2011, Hu, 2019] proclaims sharing of resources is a potential way to overcome market fluctuations. The sharing economy offer consumers flexibility to access goods and services for short term needs. Sharing helps in reducing societal cost such as production and disposal of products [Benjaafar and Hu, 2020].

[Ma et al., 2018] describe resource sharing as “the process of leveraging capabilities and assets or investing in capabilities and assets with supply chain partners". In the context of operations management(OM), sharing economy is referred to “business model built around on-demand access to products and services mediated by online platforms that match suppliers and service providers to buyers"[Benjaafar and Hu, 2020].

Further, literature review gives an overview on sharing economy around different business practices. *Sharing Economy* is a very broad concept conceding “anything and everything can be shared with proper planning and sufficient amount of collaboration between partners". [Benjaafar and Hu, 2020] shed lights on the recent sharing applications that has garnered much attention from operations management community. Author has explained business environment with key distinctive features like “*Peer-to-Peer resource sharing, On-demand service platform and On-demand service networks*", pointing out pros and cons. A connection between classical OM theory and sharing applications has been laid out stating various techniques of *Inventory Theory, Revenue Management Theory and Queuing Systems* that can still be adapted for effective matching and decision making. *Queuing Systems* techniques such as self scheduled servers, servers with probabilistic returns and double ended queue can be used to model on-demand service platform, on-demand rental networks and peer-to-peer resource sharing.

[Ma et al., 2018] developed resource sharing mechanism for sustainable production of mass customized clothes in garment industry. In the network, two manufacturers with homogeneous machines share orders among themselves during the time of insufficient capacity. Discrete event simulation study is done by developing *cost, customer satisfaction, resource waste* as key performance indicator to measure the effect. Mechanism used is very simple and only highlight benefits of sharing. [Becker and Stern, 2016] has considered network of four manufacturers, each manufacturer maintaining its own homogeneous manufacturing machine. Manufacturers are in collaboration and geographically located at different locations, working independently. Manufacturer with undercapacity release its job that can be seen by whole network. Job is allocated to manufacturer with lowest travelling distance and low machine utilization rate. After allocation, marginal return of each accomplished order is distributed equally between requesters and providers to have fair financial benefits. Since companies at BIC are located in the same vicinity, the above allocation decision-making parameters are inadequate to use. [Freitag et al., 2015] has considered a collaborative consumption business practice, where three companies have together and equally invested on two production lines. All the companies have different workload per customer order, interdeparture time and shipping time. Author modelled it as *First Come First Serve* queuing network. Companies with smaller workload have to wait in the queue, resulting in increase of mean cycle time for them.

[Xu and Yu, 2014] developed resource allocation mechanism for on-demand resource management in cloud computing. Every time user's heterogeneous requirement is submitted on cloud, the goal is to map them with idle capacity at cloud provider. Author has considered one shot allocation in this paper which is done every three hour. In cloud computing, resource requirement is comprised of $CPU core$, $memory$, $disk storage$, which is heterogeneous per requester. To achieve fair allocation author considered three properties (1)*Sharing incentive*: Amount of resource each user should receive is at least as much as simply splitting the total resources equally (2)*Envy Freeness*: No user prefers the allocation of another user (3)*Pareto Efficient*: It should be impossible to increase resource amount of a user without decreasing allocation of another user. These fairness properties can be used to develop mechanism of *Full sharing scheme* at BIC when companies have together invested in resource.

[ÃdÃ¡m Szaller et al., 2020] developed resource sharing mechanism in distributed manufacturing. A collaborative framework of manufacturing agents is introduced where members with resource shortages can request others, divide request among multiple agents, reorganize their production to be able to complete a request. Main focus was to differentiate between reliable and non-reliable participants through ratings given on past commitment and promises(successfully performing tasks, meeting due dates). Impact of mechanism is investigated through multi-agent simulation.

[Ye et al., 2017] considered fair task allocation problem in transportation. Problem is characterized in such a way that jobs to be performed are known one day ahead from terminals on which multiple trucking companies can bid depending on their idle trucks at specific times. Optimal allocation was to distribute tasks as evenly as possible among heterogeneous trucking companies who have different capacities and cost to execute tasks. Objectives were to maximize allocation, fair job distribution among bidders and minimize total compensation for job provider. Fairness objective is achieved through *max-min fairness principle*. The principle's central idea is to maximize utility for bidders. The optimization is represented through *Network Flow Games*. [Ye and Zhang, 2016] has modelled agent's participation behaviour through prospect which investigates how allocation influences agents decision to participate in a network and its effect on the systems long term social welfare. Task allocation algorithms from [Ye et al., 2017] were studied and compared through simulation. Analyses confirms that fairness motivate companies to participate in a network and eventually leads to a higher social welfare.

[Renna and Argoneto, 2011] developed conceptual model for capacity sharing in a network of independent factories. Cooperative Game Theory approach is used to facilitate coordination mechanism. Bargaining set is used as solution concept and mechanism is performed by the computation of core game. Experiments were performed considering 6-9 factories by generating fluctuating demand through uniform and normal distribution and KPI's were developed to measure the performance in multi-agent simulation environment. Results were compared with traditional negotiation model as benchmark. Although experiment consider upto 9 agents but the work is limited to single product and resource type. [Benjaafar et al., 2019] considered peer-to-peer sharing where owners are able to generate income through renting their products while no-owners are able to access these products by renting. Matching is facilitated by a mediation platform that sets the rental price and charge commission fees. Model takes a short term view over a course of a day where supply translates into individual products available for rent. An assumption is made that supply doesn't fluctuate overtime when rental requests arrive. Allocation decision is made on price by the mediator platform either to maximize its own profit or social welfare. Results show that consumers always benefit from collaborative consumption. This is because no-owners with most usage end up renting the most whereas owners with least usage end up earning through rent. Author reveals platform is most profitable when owners and rentals are sufficiently balanced so that maximum requests can be allocated.

[de Weerdt et al., 2012] proposed task allocation problem where agents are connected in a social network and tasks arrive at the agents distributed over the network. In problem setting, each agent is connected to limited number of other agents to get their task completed. The problem was to determine which resource must be assigned to which task in order to maximize the allocation. Generally, task allocation problems are NP-complete when centre can allocate tasks to every agent in the system as long as agent has required resources. Greedy algorithm was developed to solve the problem. Agent cannot use more than available resources, each allocated task must be complete, task allocation must obey social relationship are the following constraints to develop the model. Algorithm has a local knowledge about tasks and resources.

During past few years, cloud manufacturing has attracted large amount of research interest across globe. New industrial manufacturing trends and requirements such as globalisation, collaboration, integration, digitalisation etc is enabled through newly emerging technologies such as IOT, AI, CPS and big data analytics. Operational model of CMfg consist of three entities operator, provider and consumer, reveals [Liu et al., 2019]. Survey on state-of-the-art and research challenges in CMfg scheduling shown in [Liu et al., 2019] defines scheduling in narrow and broad sense. In narrow sense scheduling only refers to process of allocating resources to tasks, further monitoring status and task execution to satisfy consumer's individualised requirements. In broad sense scheduling also involves service discovery, resource capabilities, matching, selection and composition of tasks which is more practical while modelling. Therefore, further we narrow down our research towards broad sense. Usually in CMfg platform multiple companies are registered at clouds with their resources. Consumers arrived demands are either dynamically scheduled to available resources on arrival or statically planned for a day/week/month where arrived orders are collected over a period of time and scheduled in one shot. [Zhou et al., 2018a] has developed event-triggered dynamic scheduling method for allocating randomly arriving tasks to machines. Author proposed decomposing each task into number of subtasks, since not each company has available machines that can accomplish whole task. Registered companies at CMfg platform are located at different location. Therefore, combination of resource service time, logistic time and earliest available time of resources at candidate companies are considered for allocation to reduce overall makespan of each task as objective. [Elgendy et al., 2019] developed MILP model for collaborating distributed manufacturing capabilities. Tasks are allocated to resources with an objective to reduce overall makespan for each task. Author has also integrated transportation time if job has to travel from one company to another or its sequential operations. Further to accommodate large size problem instance Genetic Algorithm is developed which is integrated with event-triggered strategies in order to improve the efficiency of GA. [Li et al., 2018] proposed multi-agent based approach to achieve global optimal schedule. Scheduling architecture is developed which is solved through two heuristics algorithm i.e. MAS-MSDT and MAS-LSS. In MAS-MSDT, Shortest-delivery-Time rule is followed. Job with shortest delivery time is computed, then machine with the smallest makespan is selected to schedule the job. In MAS-LSS, for each machine, first job is assigned that has shortest delivery time and then job with longest delivery time is assigned to the same machine.

Most of the literature in CMfg make scheduling decision to reduce overall makespan for each task that select resource contingent to earliest available time, process time and logistics time. But there can also be different evaluation parameters with non-identical dimensions for allocation like quality of service, varied cost for different resources provided by distinct companies. [Cao et al., 2015] and [Zhou et al., 2018b] proposed service selection rule and scheduling strategy incorporating cost, quality and time for scheduling. In [Cao et al., 2015] fuzzy decision making theory is adopted to transform multi-dimensional objective evaluation indicators into uni-dimensional to remove non-standardization errors. In [Zhou et al., 2018b] min-max normalization method is used to reconcile the conflict among multi-dimensional objective evaluation indicators. Quality is considered as trust value in [Cao et al., 2015, Zhou et al., 2018b], which is incorporated in decision making for allocation. Experiments performed in these literature consider random trust value based on past performances of companies for a horizon and results manifest that companies with low resource cost and high trust value has better chance of getting tasks allocated to their resources.

Trust value is a fundamental concern in large scale open distributed systems where companies need to trust each other for sharing movable physical resources or getting their tasks accomplished on immovable resource in uncertain and constantly changing environment. [Ramchurn et al., 2004] conceptualise trust value at individual-level and system-level. At individual level tenant has some belief about the honesty of its interaction partner, whereas at system level tenants are forced to be trustworthy by the rules of encounter(i.e. mechanism or protocol) that regulate the system. Author shade lights on reputation mechanism at both the levels. At individual level, each tenant maintain their own metrics of rating based on the past interaction with other tenants. At system level, tenants provide truthful ratings to other tenants after interaction which is then aggregated at system level and defines the individual trust value. To make reputation model practically applicable there are three underlined aspects: 1. devising method to gather rating 2.

devising reliable reasoning methods to rate 3. devising mechanism to promote rating that describes trustworthiness of company[Ramchurn et al., 2004]. [Xu and Yu, 2014] annotates transaction duration, product quality, financial transaction and service level as reasons to provide rating. For each transaction, trust is evaluated as sum of indicators multiplied with weight factors to obtain single rating given by consumer to provider.

2.2 Motivation

Literature survey provides a holistic view on resource sharing under diverse business practices. It is possible to extrapolate some useful insights from the literature work. After literature survey we can say that resource sharing can be facilitated in various ways. For example, multi-agent systems or game theory approach can be considered to extrapolate the effects of agents cooperative and non-cooperative behaviour through simulation study. Simple or complex allocation decision making model can be developed formalizing objective function as utility/nearby geographic location/ makespan/ trust value/ cost. Objective function for model can be formulated with single objective or combination of multiple objectives. Research done by [Freitag et al., 2015, Ma et al., 2018, Becker and Stern, 2016] focus on showing the benefits of resource sharing, manifesting improvements through key logistical figures but hardly explain concept of detailed sharing mechanism that can be practically applicable. Their research is only limited to homogeneous resource sharing amongst 2-3 companies. Practical assumptions like machine capabilities, available machine time windows, tasks with multiple machining process are missing. [Elgendy et al., 2019, Zhou et al., 2018a, Liu et al., 2019, Cao et al., 2015] consider CMfg platform for establishing distributed manufacturing environment operated via central cloud. Generally in CMfg, manufacturers register their resources to central cloud and customers arriving requests on central cloud are scheduled to manufacturers, where it is always contemplated that for each horizon machines are fully available. But in our case companies schedule their own customer orders to their machines and leftover available machine times are considered to be shared at cloud. Therefore, for each horizon, resource availability at each company for each machine can be heterogeneous allied to time with diverse time windows. We refer above papers and seize ideas on CMfg, multiple tenants, heterogeneous resources, Job shop scheduling problem, Resource constrained project scheduling problem to develop optimization model to share resources at BIC.

Generally, given an option, consumers are interested in buying a product at cheaper price from trustworthy provider. When sharing resource between finite set of agents, social welfare is a need to have long term cooperation. The mediator platform must ensure that the allocation is unbiased. Idea developed in this research is highly motivated by E-commerce platform. For example, consumer needs to hire rental car for short period of time but do not want to own it. Given multiple options, (1) provider with low rental cost and high quality service level (2) provider with high rental cost and high quality service level (3) provider with low rental cost and low quality service level (4) provider with high rental cost and low quality service level, consumer would go for provider who has better quality rental service based on past interaction with other customers (customer reviews) and cheaper price. We hypothesize that provider with option (1) has highest probability of getting maximum bookings, whereas provider with option (4) has lowest probability. We develop a framework to show how resource can be practically regulated considering cost and trust factor as objective function to make scheduling decision. In almost all related literature work referred in section 2.1, it is always considered that resources are available from start to end of horizon working 24/7 everyday which is impractical. There can be breaks, unavailable workforce, machine breakdowns etc. To overcome this issue we introduce multiple time windows at machines available with different providers. This way providers have flexibility to only release available machine up-times on cloud which our model will consider to generate schedules. Authors have shown effect of trust value on scheduling. However, trust value update mechanism is missing which is an important factor when scheduling of resources for the next horizon needs to be considered allied to past performances of providers. Therefore, we introduce an update mechanism for trust value. Optimization model developed in this paper is beneficial to research community of cloud manufacturing and flexible job shop scheduling problem.

Chapter 3

Problem Description

3.1 Problem context

Tenants at BIC would be working independently with their own set of resources in a decentralized way. Each tenant has their own production planning for received customer orders. Depending on customer order size and available resource capacity each tenant decide its production move rate to accomplish orders under due date. If there is high demand, tenants face issue of resource undercapacity. To tackle this situation tenant has to either ask customer to be flexible and increase due date, otherwise orders are considered as lost orders. On the other hand, if there is low market demand, tenants face issue of resource overcapacity which leads to excess resources. Due to fluctuating individualized customer demand, tenants can either be in above two situation. Tenants manufacturing individualized product in high-mix,low-volume market are heavily dependent on demand forecast. Demand prediction is more accurate when it is closer to actual demand. Investing on resources or infrastructure depending on mid-term or long-term planning might go wrong due to unexpected short term demand discrepancy. Here at BIC, tenants would have a flexibility to efficiently plan the resources and complete customer orders through sharing of resources even though prediction does not befall as expected. Since, tenants cannot predict availability of resources in future, we consider one day prior to actual production day tenants have clear idea on amount of resource they require to produce their own customer orders for next day. Therefore, we consider planning horizon for sharing as one day. Firstly, if tenant has resources just enough for its own production, it doesn't participate in sharing. Secondly, if tenant has extra resources than required it becomes *Provider*. If tenant has customer orders beyond resource capacity and hence require extra resources, it becomes *Requester*. This way everyday tenants would participate in sharing as per their need.

3.2 Problem Definition

Resource sharing problem studied in this paper deals with CNC machine as a manufacturing resource to be shared. The shop floor environment being studied is shared amongst multiple tenants. The set C = $\{1, 2, .., |C|\}$ is a set of companies participating in sharing. Set of companies providing machines are providers P = $\{1, 2, .., |P|\}$, where $P \subseteq C$. Set of companies requesting for machines are requesters R = $\{1, 2, .., |R|\}$, where $R \subseteq C$. All the companies are connected to central cloud. Sharing of resources is facilitated on daily basis, therefore planning horizon for sharing is one day, represented by H_n where n denotes horizon (i.e. day). To facilitate sharing, we consider all the requesters to formulate its requirement and redirect it to the central cloud. Requirement formulated by requester is further defined as task denoted by i, where $N = \{1, 2, .., |N|\}$ is set of initiated tasks. Each task has a job (raw material provided by requester) that may need to go through single machining operation to multiple operations as per requirement of requester. Each company at BIC can be heterogeneous with respect to number of machines and capability of each machine to perform all requested machining processes from requester for each task. It can be possible that one or multiple machines at each provider are not adequate to perform all the machining processes and complete the task, however, multiple companies together may be able to accomplish the task. Therefore, we divide each task into number of subtasks where each subtask is independent machining operation. Subtask is denoted by j, where $Z_i = \{Z_1, Z_2, .., |Z_i|\}$ is a set of subtasks in each task i. For example, let us consider a task with job that needs to go through [milling, drilling, boring], this task would be divided into subtasks as [“a”: milling, “b”: drilling, “c”: boring]. All tasks are independent of each other, but there is precedence relationship between subtasks of task. Subtask “b" cannot start until “a” is finished and “c" cannot start until “b" is finished.

Shop floor can be a blend of small, medium or large enterprises, therefore, machines owned by each company may vary in number. Every machine on the shop floor has its own unique id that is denoted by $k \in M$, where M is the set of all machines at shop floor. M_R and M_P represents set of machines owned by requesters and providers. Machines of each company may be diverse due to its own business practice. Therefore, capabilities of machines owned by each company may be heterogeneous. Capability of machine is defined as carrying out different kind of subtask j (i.e. machining operations) in task i, where A_k represents set of capabilities of machine k (See Appendix A to check capability of each machine with its unique id and owner company). We consider that all the machines are registered at central cloud with its capabilities. For each planning horizon, provider reserves particular number of time windows on each machine to carry out its own production which are considered as busy time windows. However, remaining idle time windows released by providers on cloud can be used for scheduling requesters tasks. There can be single to multiple available time windows on each machine denoted by $e = \{1, 2, .., |e|\}$, where $w^e_{k_l}$ denote lower bound of available time window e at machine k and $w^e_{k_U}$ denote upper bound of available time window e at machine k.

Every subtask of a task has some processing time t_{ij} which comes with an initiated task from requesters side. We consider that for each subtask if machine capability is matched with any of the providers machine on shop floor, processing time would be same. For each planning horizon, provider has an opportunity to release hourly rate r_k in euros for each machine k, to gain profit through sharing. For getting each subtask of a task accomplished on providers machine, requester has to pay $cost^{RP}_{ij} = t_{ij} \times r_k$ in euros.

We know that CNC machine is an immovable resource. Therefore, for each task with more than one subtask, job may have to be moved from one machine to another for subsequent subtasks comply to its schedule. There would be some travelling time TT^{ijk}_{ij+1l} in minutes for moving job from machine k to l for processing subtask $j + 1$ after subtask j is finished on machine k from task i. Machine being immovable resource type, travelling time is a parameter that has fixed value (Matrix of travelling time between machines is provided in appendix A). Travelling time is translated to travel cost TC^{ijk}_{ij+1l} as 1 cent per second of travel. Travelling cost increases with increase in travelling time between machines. We consider soft deadline from requester’s side, where requester expects to get its task accomplished by end of the day. Every subtask of a task has start time s_{ijk} and completion time c_{ijk} on machine.

After scheduling is done, all the scheduled subtasks of tasks are completed by end of the horizon and requesters have to rate provider on whose machine subtasks were scheduled. There is a possibility that subtasks of same task might get scheduled on different machine of distinct providers. Therefore, we consider ratings to be given on subtask level to measure service level satisfaction. Rating scales are widely used in online platform in an attempt to provide indications of consumer opinions on product. Rating scales are classified into numerical rating scale, verbal analogue scale, visual analogue scale, likert and graphical rating scale. Common sites which employ numerical rating scales are IMDB, Amazon.com, BoardGameGeek etc. [Zhou et al., 2018a] and [Cao et al., 2015] has considered numerical rating scale between 0-10 as ratings from customer to manufacturer in cloud manufacturing platform. In our work, we also consider rating value as whole number between [0,10] denoted as $rate^{RP}_{ij}$ provided by requester R to provider P for subtask j in task i. f_C represents trust value of each company.

3.3 Problem Characteristics

1. All tasks arrive at cloud platform over a period of one day, pre-emption is not allowed.
2. Each machine can only perform one subtask of a task at a particular time.
3. Each subtask can be performed by only one machine , i.e. no duplicate allocation.

4. Precedence relationship exists only between subtasks that belong to the same task.
5. Travelling time for moving job from machine-to-machine for distinct subtasks of a task.
6. Each machine may have heterogeneous available time windows.
7. Machine may have multiple available time windows that are released by providers on central cloud.

3.4 Conceptual Framework

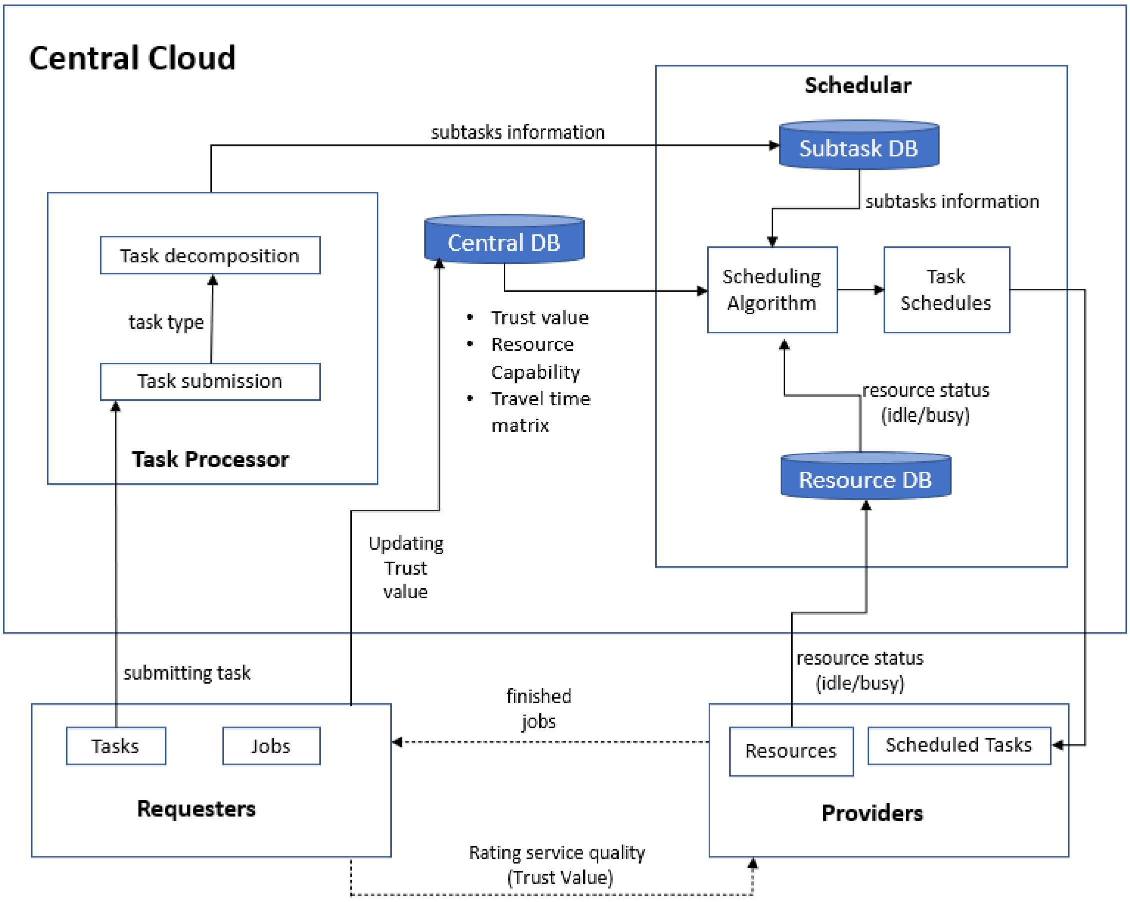

Figure 3.1: Framework

Above framework gives conceptual idea of how resource sharing can be facilitated at BIC. Base framework is adapted from [Zhou et al., 2018a] and further developed as per our problem definition. There are some prerequisite to enable sharing, (1) All companies must register their machine at central cloud with the capability of machining process (subtask) each machine can handle (2) Cloud must have information on fixed position of each machine at the shop floor (3) Cloud must have information of updated trust value of each company. One day prior of sharing, requesters formulate their requirements and send it to central cloud over a period of day. Once all the tasks are collected at central cloud, task processor identify each task and decompose it into multiple independent subtasks as per data structure required as input to model. After processing, it is submitted to Subtask DB(database) at scheduler. Next, companies are required to send the machine status for planning horizon i.e. 24 hours. Status of each acquired machine is converted into data format as required and stored in Resource DB(database). Once we have all the data of arrived tasks as well as machine status, we run our optimization model (scheduling algorithm) developed in section 4.1 to schedule subtasks of tasks to available machines. Once whole task is completed requesters provide rating on quality of service provided by providers. As soon as all the ratings are received by the end of the day, trust values of each company is updated at Central DB(database) by mechanism developed in section 4.2. Central DB holds all relevant information about machine capabilities, updated trust values, machine-to-machine travelling time. This is the whole idea to enable sharing. In this research work, we develop scheduling algorithm and trust value update mechanism formulated in chapter 4.

Chapter 4

Mathematical Formulation

In this chapter, we develop optimization model and trust value update mechanism of the problem specified in problem description.

4.1 Optimization Model

Decision variables of the model that needs to be optimized are:

Binary variables

x_{ijk} 1, if subtask j of task i is scheduled to machine k: 0, otherwise

y_{ijhgk} 1, if subtask j of task i precedes subtask g of task h on machine k: 0, otherwise

xx^{ijk}_{ij+1l} 1, after processing subtask j of task i on machine k if job need to be transferred to machine l for processing next subtask $j+1$

v_i 1, if all the subtasks j of task i are scheduled: 0, otherwise

Continuous variables

s_{ijk} starting time of subtask j of task i on machine k

c_{ijk} completion time of subtask j of task i on machine k

q^{ijk}_{ij+1l} variable to store travel time each time job of task i has travelled from machine k to l for next subtask $j+1$

Integer variable

u_i variable to store total number of scheduled subtasks j in task i

Objective and constraints

As a requester, one would always prefer a provider who provides better quality machine at the cheapest cost. As a provider, one would like to earn some profit by providing machines. We assume hourly rate of machine released by each provider to share its machine on cloud would gain them some profit. Before we explain optimization model, there can be two scenario's for each fixed planning horizon that needs to be highlighted, (1) Overall hours required to accomplish all the tasks are less than overall available hours on machines, (2) Overall hours required to accomplish all the tasks are more than overall available hours on machines. In scenario (1), there can be lot of options to schedule each subtask of task on different machine at distinct provider with varying cost and trust value. Due to surplus available option, we know all the subtasks of tasks can be scheduled to machines. In scenario (2), due to huge number of tasks and less available machines not all the tasks can be completed. We want our model to handle both the situations explained in scenario (1) and (2).

Objective function of the model is defined as weighted summation of cost and trust factor. For each subtask of a task, we need to search for multiple scheduling-solution to optimize the objective function. For each planning horizon, we want our objective function to complete maximum number tasks at cheapest cost with machines of trustworthy providers. The objective function is formalized in equation (4.1) and constraints from (4.2)-(4.17)

Optimization Model

Objective:

$$max \quad \alpha \sum_{i \in N} \sum_{j \in Z_i} \sum_{k \in M} \frac{t_{ij}.x_{ijk}.v_i}{r_k} + (1-\alpha) \sum_{i \in N} \sum_{j \in Z_i} \sum_{k \in M} f_C.x_{ijk}.v_i \tag{4.1}$$

Subject to:

$$\sum_{k \in M} x_{ijk} \leq 1 \qquad \forall \quad i \in N, \quad j \in Z_i, \quad j \in A_k, \quad R_i \notin M_R \tag{4.2}$$

$$\sum_{k \in M} x_{ijk} = 0 \qquad \forall \quad i \in N, \quad j \in Z_i, \quad j \notin A_k \tag{4.3}$$

$$u_i = \sum_{j \in Z_i} \sum_{k \in M} x_{ijk} \qquad \forall \quad i \in N \tag{4.4}$$

$$v_i = \frac{u_i}{b_i} \qquad \forall \quad i \in N \tag{4.5}$$

$$\sum_{k \in M} t_{ij}.x_{ijk} \leq a_k \qquad \forall \quad i \in N, \quad j \in Z_i \tag{4.6}$$

$$w^e_{k_l}.x_{ijk} \leq s_{ijk} \qquad \forall \quad i \in N, \quad j \in Z_i, \quad k \in M, \quad e \in e_k \tag{4.7}$$

$$c_{ijk} \geq w^e_{k_u}.x_{ijk} \qquad \forall \quad i \in N, \quad j \in Z_i, \quad k \in M, \quad e \in e_k \tag{4.8}$$

$$s_{ijk} + c_{ijk} \leq x_{ijk}.L \qquad \forall \quad i \in N, \quad j \in Z_i, \quad k \in M \tag{4.9}$$

$$xx^{ijk}_{ij+1l} = x_{ijk}.x_{ij+1l} \qquad \forall \quad i \in N, \quad j \in Z_i, \quad k,l \in M \tag{4.10}$$

$$c_{ijk} \geq s_{ijk} + t_{ij} + TT^{ijk}_{ij+1l}.xx^{ijk}_{ij+1l} - (1 - x_{ijk}).L \qquad \forall \quad i \in N, \quad j \in Z_i, \quad k,l \in M \tag{4.11}$$

$$c_{ij_nk} \geq s_{ij_nk} + t_{ij_n} - (1 - x_{ij_nk}).L \qquad \forall \quad i \in N, \quad j \in Z_i, \quad k \in M \tag{4.12}$$

$$q_{ij+1l}^{ijk} = xx_{ij+1l}^{ijk}.TT_{ij+1l}^{ijk} \qquad \forall \quad i \in N, \quad j \in Z_i, \quad k,l \in M \tag{4.13}$$

$$s_{ijk} \geq c_{hgk} - (1 - y_{ijhgk}).L \qquad \forall \quad i,h \in N, i < h, j \in Z_i, g \in Z_h, k \in M \tag{4.14}$$

$$s_{hgk} \geq c_{ijk} - y_{ijhgk}.L \qquad \forall \quad i,h \in N, i < h, j \in Z_i, g \in Z_h, k \in M \tag{4.15}$$

$$\sum_{k \in M} s_{ij+1k} \geq \sum_{k \in M} c_{ijk} \qquad \forall \quad i \in N, \quad j \in Z_i \tag{4.16}$$

$$s_{i,j,k} \geq 0 \qquad \forall \quad i \in N, \quad j \in Z_i, \quad k \in M \tag{4.17}$$

$$c_{i,j,k} \geq 0 \qquad \forall \quad i \in N, \quad j \in Z_i, \quad k \in M \tag{4.18}$$

In objective function (4.1), α is a weight factor. In fraction $\frac{t_{ij}.x_{ijk}.v_i}{r_k}$, numerator try to schedule maximum number tasks, where v_i is a binary variable defined in constraint (4.5) that gives value 1 each time all subtasks of a task are scheduled and denominator has hourly rate r_k of machine. Together whole fraction try to schedule maximum number of subtasks of tasks to the cheapest machines of providers to reduce overall cost of each task. In $f_C.x_{ijk}.v_i$, f_C is the trust value of company where equation try to schedule maximum subtasks of tasks to company with highest trust value. Constraint (4.2) ensures that each subtask must be scheduled to capable machine only once i.e. no subtask must be repeatedly scheduled and requester who has initiated the task on central cloud, the task should not be scheduled to the set of machines provided by itself. Constraint (4.3) guarantee that no subtask must be scheduled if machine's capability does not match. In constraint (4.4), variable u_i is used to store total number of scheduled subtasks of each task. In constraint (4.5), variable v_i is defined that store binary value 1 if all the subtasks of each task are scheduled. Constraint (4.6) ensures that total hours used in scheduling should not exceed total available hours on each machine. Constraint (4.7) makes sure that subtasks should always be scheduled from lower bound of time window e, whereas constraint (4.8) ensure it does not exceed upper bound of time window e. Together, constraints (4.7)-(4.8) ensure that all the subtasks are scheduled in between available time windows. If subtask j is not assigned to machine k, constraint (4.9) set its starting and completion time on machine k equal to zero. Constant big number L is large enough to ensure correctness of the constraint. After scheduling subtask j of task i to machine k, then next subtask $j + 1$ of same task i is scheduled to which machine l is defined in constraint (4.10) that gets binary value 1. Constraint (4.11) define that completion time of each subtask is always greater than or equal to sum of its start time, process time and travelling time of job for next subtask in a task. When last subtask of each task is scheduled, job does not have to travel ahead and we know that overall task is completed. Therefore, constraint (4.12) ensures that for last subtask of each task completion time is sum of start time and process time. In constraint (4.13), we define variable q_{ij+1l}^{ijk} to store travel time if job has travelled from machine k to machine l for distinct subtasks of each task. Constraints (4.14) and (4.15) together ensures that no subtask must overlap on any machine while scheduling. Constraint (4.16) ensures that the precedence relationship between subtasks of task are not violated, i.e. subtask $j + 1$ cannot start until j is finished for task i. Constraints (4.17) and (4.18) ensures start time and end time cannot have a negative value.

We now use the following small example to illustrate the scheduling of task to machines through optimization model.

Example 1. Suppose for planning horizon H_n, 4 tasks are submitted to central cloud. Companies 1 and 4 are requesters that has initiated tasks as per their requirement. Each task has single or multiple subtasks.

We consider 4 tasks with heterogeneous subtasks whose processing time is defined in Table 4.1. For this example, one machine at each provider is considered that are available at distinct time with multiple time windows as shown in Table 4.2. Companies 2,3,5 and 6 are providers. To see the difference in scheduling we have set same capability to all the machines. Data is defined is such a way that company-2 has high machine hourly rate and high trust value. Company-3 has high machine hourly rate and low trust value. Company-5 has low machine hourly rate and low trust value, whereas company-6 has low machine hourly rate and high trust value. All the input data on behalf of requesters and providers side is defined in Table 4.1 and 4.2

Table 4.1: Requesters input data

Requesters			
Company	Task	Subtasks	Process time
1	1	a,c,d,f	2,3,4,2
1	2	a,b,d	3,3,4
4	3	a,c,e	1,4,5
4	4	f	4

Table 4.2: Providers input data

Providers					
Company	Machine id	Capability	Available time Windows	Hourly rate	Trust value
2	9	a,b,c,d,e,f	(7,13),(16,22)	730	9.2
3	15	a,b,c,d,e,f	(2,8),(15,18),(20,23)	720	6.5
5	25	a,b,c,d,e,f	(10,22)	550	7
6	29	a,b,c,d,e,f	(5,17)	580	9.4

As we have all the input data, we use optimization model to schedule subtasks on machines. We run our model thrice allied to cost factor, trust factor and equal weighted objective function in order to investigate the results. Scheduling results are distinguished in Figure 4.1, 4.2 and 4.3. Y-axis of Gannt chart has machine id with its provider company, for example Machine15-C3(machine 15 belongs to company 3). On X-axis we have horizon of one day(24 hours). In Gannt chart, blue bars indicate busy time windows that provider has reserved for themselves to perform its own operations, but time windows that are idle can be used to schedule requesters tasks. It can be seen in Figure 4.1 that model has only considered idle time windows to schedule. Multi-coloured bars are distinct tasks comprised of multiple subtasks. For example, orange colour bar is Task01(T01), where T01-a,T01-c,T01-d,T01-f are its subsequent subtasks. All the subtasks are scheduled because of excess machines availability. Empty spaces in between bars indicate excess available idle machine time which were not utilized for scheduling.

Figure 4.1 manifest scheduling result when decision is made on cost factor. Objective function try to find the best optimal scheduling solution so that overall cost of tasks are minimized. Machine25-C5 and Machine29-C6 are fully utilized as hourly rate is low, then Machine15-C3 and Machine09-C2 were considered for scheduling. Figure 4.2 manifest scheduling result when decision is made on trust factor. Since, provider 2 and 6 has high trust value, model tried to prioritize their machine and then untrustworthy providers were considered. Next, we give equal weightage to both cost and trust factor in the objective function. It can be seen in Figure 4.3, maximum number of subtasks were scheduled to Machine29-C6 because it has low cost and high trust value. Machine09-C2 and Machine25-C5 were preferred equally infact model tried to make a trade off because provider 2 has high hourly rate but it has a high trust value, whereas provider 6 has low trust value but due to low hourly rate got subtasks scheduled on its machine. Machine15-C3 just got 1 subtask because there was no option left. Provider with low trust value and high hourly rate was least preferred.

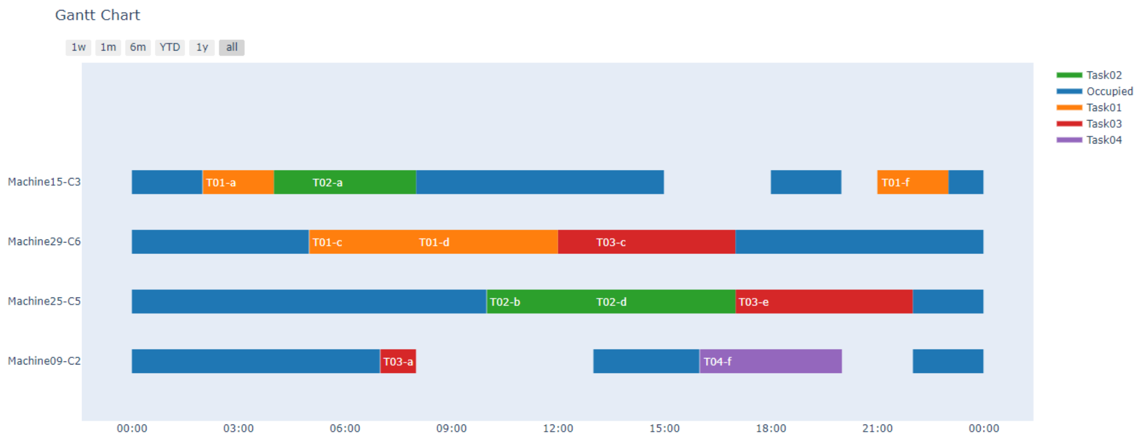

Figure 4.1: Scheduling allied to cost factor

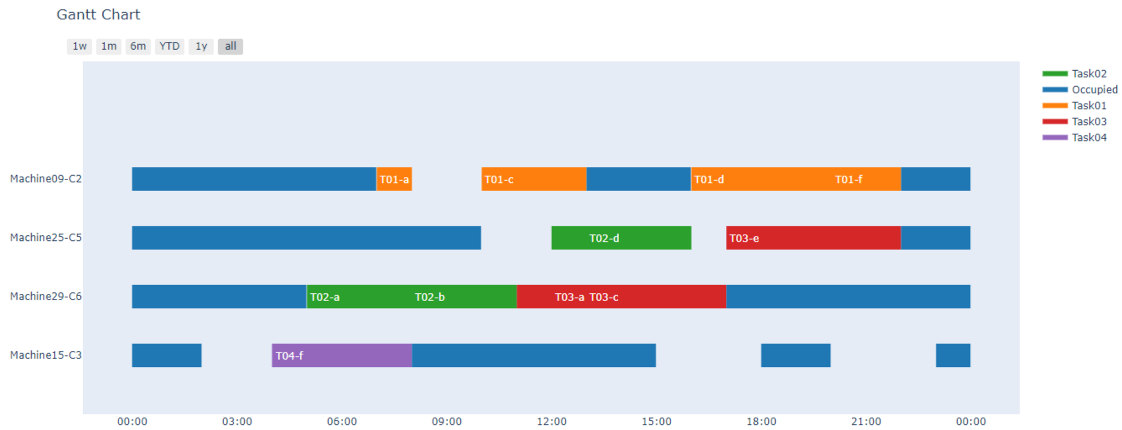

Figure 4.2: Scheduling allied to trust factor

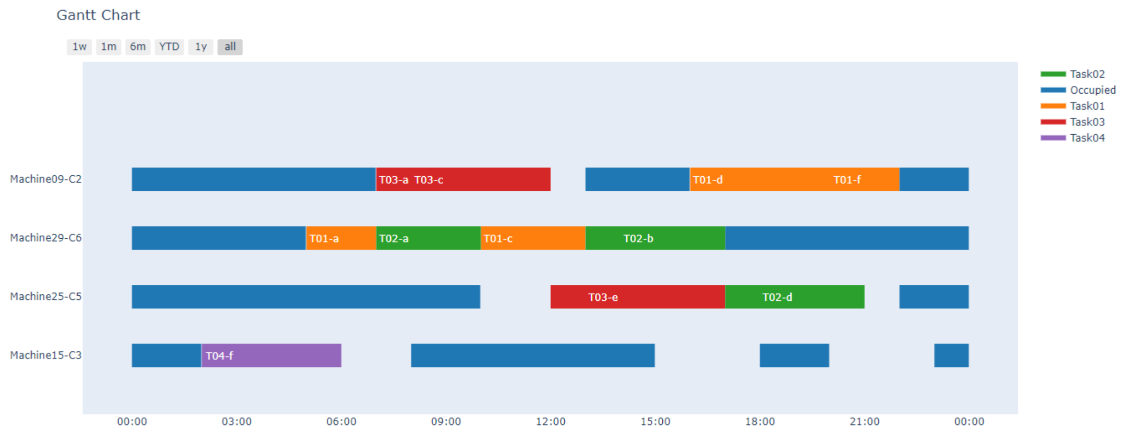

Figure 4.3: Scheduling allied to equal weightage of trust and cost factor

4.2 Trust value update mechanism

After all the scheduled subtasks of tasks are accomplished for a horizon. It is important to update the performance of each provider with the ratings provided by requesters so that updated trust value can be further used for scheduling in the subsequent horizon. To make scheduling decision for current horizon H_n, updated trust value of previous horizon H_{n-1} is used. Process flow chart in Figure 4.4 emphasize the mechanism.

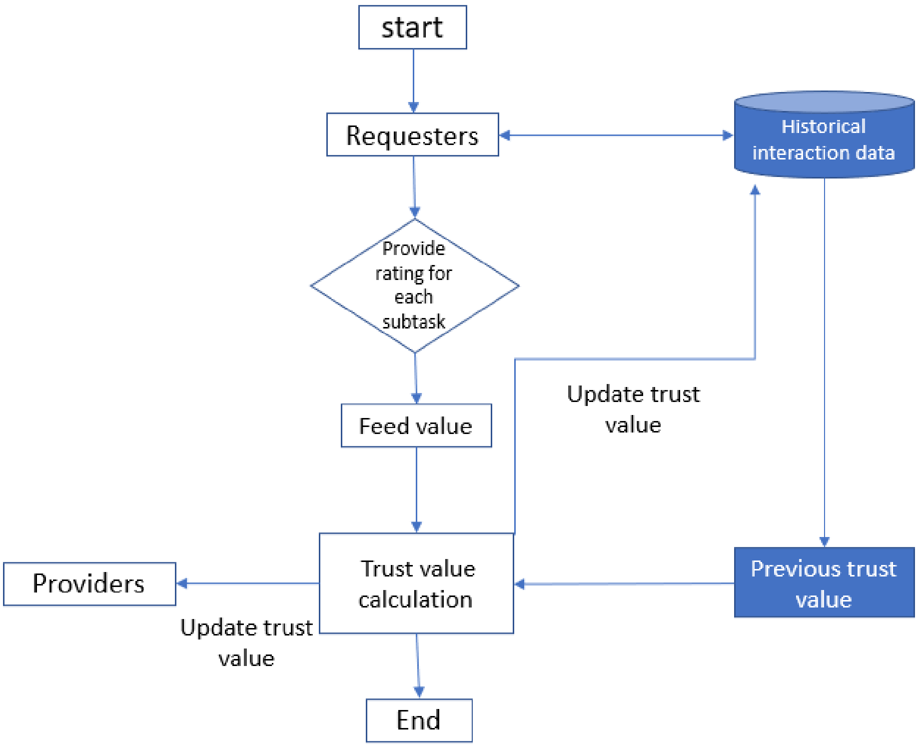

Figure 4.4: Update mechanism

All the ratings of requesters to providers by the end of current planned horizon H_n as well as updated trust value and all the scheduled subtasks until previous horizon H_{n-1} are the input data to evaluate the trust value of each provider which is calculated as cumulative moving average (CMA). Main motive behind using this particular formula is its potential to consider all past interactions of previous horizons between companies to update trust value every time sharing is facilitated in the next upcoming horizon. This way company who performed really well in the previous horizons does not loose its credibility in the environment due to bad performance in few horizons. At the same time, company who has been performing bad in the previous horizons needs to regain its trust value by performing well in upcoming horizons. Formula to calculate CMA is formalized in equation (4.19):

$$f_C = CMA^P_{H_n} = \frac{\sum_{j=1}^{m_P} rate^{RP}_{ij} + n_P \times CMA^P_{H_{n-1}}}{n_P + m_P} \tag{4.19}$$

It is instinctive that if j number of subtasks are scheduled to machines of provider, it will receive same number of distinct ratings from requesters. Therefore, cumulative average of each provider is summation of ratings for all the received subtasks in H_n plus product of total scheduled subtask until previous horizon times provider's previous trust value divided by sum of previously total scheduled subtasks n_P until H_{n-1} and newly scheduled subtasks m_P in H_n. In table 4.3, values in all multi-column of H_{n-1} are assumed

H_{n-1}			H_n		
Company	Trust value	Total scheduled subtasks	Company	Trust value	Total scheduled subtasks
1	7.5	110	1	7.5	110
2	9.2	149	2	9.16	153
3	6.5	58	3	6.52	59
4	8.3	63	4	8.3	63
5	7	189	5	6.98	191
6	9.4	58	6	9.425	62

Table 4.3: Trust value update

values just to show in what way previous 'Trust Value' is considered for scheduling of H_n. To calculate 'Total scheduled subtasks', we have developed a database of excel file in python. Each time optimization model is run, total number of scheduled subtasks to set of machines at each provider M_P in each horizon is fetched through output of decision variable x_{ijk} and updated in excel file. We continue with example 1 to quantitatively manifest the calculation.

From example 1, we consider scheduling results of equal weightage when $\alpha = 0.5$ to update the trust value. Four subtasks are scheduled to machine of C2, i.e. for all four subtasks C2 would receive some rating from requesters. Let us consider that for T03-a and T03-c, requester 4 is satisfied with the quality of service and provide with rate 10. But for T01-d and T01-f, requester 1 received machine with some glitch in tool and had to compromise with finishing of product. Therefore, C2 received lower rating for T01-d and T01-f as 6. After receiving all the feedbacks(rating), cumulative average of C2 is calculated as shown in expression (4.20).

$$\begin{aligned} CMA_T^2 &= \frac{(rate_{3,1}^{3,2} + rate_{3,2}^{3,2} + rate_{1,3}^{1,2} + rate_{1,4}^{1,2}) + n_P \times CMA_{T-1}^2}{n_P + m_P} \\ &= \frac{(10 + 10 + 6 + 6) + 149 \times 9.2}{149 + 4} \\ &= 9.157 \end{aligned} \tag{4.20}$$

Similarly, trust values are calculated for all the providers and updated at Table 4.3 in column 'Trust value'. Calculated cumulative average of each provider denote new trust value for them which is fetched as input for scheduling in next horizon.

Chapter 5

Experiment

In this chapter, we investigate the performance of proposed optimization model for sharing resources and fetch some managerial insights through numerical experiments. First of all we are interested to know, (Q.1) The amount of computational time model requires to generate solution ? Since, we have introduced trust factor with cost factor in the objective function. We are interested to know, (Q.2) How much extra cost requesters have to pay for getting their tasks scheduled to trustworthy providers? (Q.3) How much profit can companies gain through sharing ? We are interested to know, (Q.4) Is sharing better between homogeneous or heterogeneous companies sharing the platform? (Q.5) Which company would earn maximum profit ?

We generate various test instances with different parameters and compute solutions to answer the questions. Optimization model is coded in python language using Gurobi python package to optimize the objective function. All the experiments are performed on 8th generation Intel Core-i5 CPU with 8 cores and 8GB RAM on Windows 10 operating system. Gurobi use Branch and bound algorithm to solve the problem.

We derive test instance from our motivating example 1. We increase our data size to make the experiments representative of the situation at BIC. In all the experiments we make use of data instances created as per information provided in Table 5.1. Number of machines owned by each company and its capabilities has been provided in Appendix A. In total, thirty number of machines are divided amongst 6 companies whose capabilities are fixed. Travelling time matrix between machines has also been provided in Appendix A. Experiments are performed using objective function derived in Equation 4.1. Further in experiments, we set weight parameter $\alpha = 1$ when scheduling is done on cost factor, $\alpha = 0$ when scheduling is done on trust factor and $\alpha = 0.5$ when scheduling is done with equal weightage of cost and trust factor. For each planning horizon, we consider that each company can be requesters as well as providers of machines.

Experimental data information	
Tasks	[1,40]
Subtasks	[1,6]
Machines	[1,2,..,30]
Companies	[1,2,..,6]
Machine hourly rate	[500,750] euros/hour
Subtask process time	[1,7] in hours
Company Trust value	[0,10]
Machine availability	[0,24] in hours
Time windows	[1,3]

Table 5.1: Input data

5.1 Computation time

It is important to validate computation time model take to generate solution. Therefore, we perform experiments with varied number of tasks and its subtasks in range, N = $\{3, 5, 8, 10, 12, 15, 18, 20, 25, 30, 35, 40\}$ and J = $\{11, 17, 23, 26, 31, 36, 45, 51, 61, 71, 86, 100\}$. In Table 5.2, column 'Cncl (N)' represents total number of cancelled tasks, whereas column 'Cncl (J)' represents total number of cancelled subtasks. At each instance we increase number of tasks and its subtasks. Total size of 30 machine with 50% availability at each machine with diverse time windows are considered to generate test instances. Moderately heterogeneous capabilities(See Appendix A) of machines are considered for the experiments performed in this section. As we run our model on gurobi, incumbent solution found is the first feasible solution through the search. An incumbent solution is the best integer solution found at any point in the search. As search carry on with time, every time new incumbent has been found is a better solution than previous one. The column '1st Incumbent sol. time' reports the time for optimization model to find the 1st feasible solution. 'Cost' is the amount of overall price paid by requesters to providers for providing machines.

From Table 5.2, it can be seen that for small instances (N = 3 to 12), optimal solution is found within few minutes. We report time to find the optimal solution as 'Comp. time'. As size of problem instance increases, the program takes potentially lot more time to reach the optimal value. Therefore, we set upper limit to 1800 seconds and the best incumbent solution is reported if the time limit is reached. During program run Gurobi optimizer show a gap between incumbent solution and optimal solution. Solution gap reduces as program search continue with time. For the last two instances solution gap was very huge between 35-40 % when time limit was set to 1800 seconds where optimal solution was far away from incumbent solution found in 1800 seconds. Therefore, we increased the time to 3600 seconds for last two instances to get better solution. If we compare scheduling results of each instance, it can be seen that requesters has to pay some extra cost for getting their tasks allocated to trustworthy provider. We calculate gap in % to show extra amount incurred by integrating trust factor by $\frac{cost(Equalweight) - cost(costfactor)}{cost(costfactor)}\%$, being positive. Since, available capacity was fixed to 50% for all the test instances. Last two instances with 35 and 40 tasks has less overall available hours on machine than required. Therefore, few tasks and its subtasks got cancelled during program run. From results in Table 5.2, it can also be seen that for last two instances gap is very small that is because of not having many scheduling options and model tried to schedule maximum number of tasks to the available machines. Figure 5.1 highlights cost to be paid by requesters in each instance and Figure 5.2 highlights time to obtain first feasible solution.

Looking at results in Table 5.2, we see that requesters have to pay 3.66 % extra cost on average when trust factor is integrated with cost. For problems (N > 12, J > 31), optimal solution is found in few minutes of run. For bigger problems (N > 30, J > 71), optimizer might take more than 1 hour to find an optimal/better feasible solution. This answers our (Q.1)

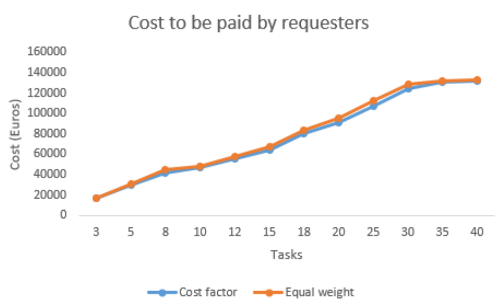

Figure 5.1: Cost paid by requesters in each instance

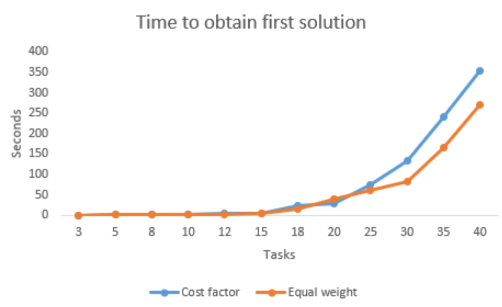

Figure 5.2: Time to obtain 1st feasible solution in each instance

Table 5.2: Computational time

				Cost factor ($\alpha = 1$)			Equal weightage ($\alpha = 0.5$)			
N	J	Cncl (N)	Cncl (J)	Comp. time (sec)	1st Incum-bent sol. time (sec)	Cost (euros)	Comp. time (sec)	1st Incum-bent sol. time (sec)	Cost (euros)	Gap (%)
3	11	0	0	2.2	0	17187	1.2	0	17462	1.60%
5	17	0	0	8.4	1	30335	5.7	1	31088	2.48%
8	23	0	0	91.9	2	41412	16.3	2	44801	8.18%
10	26	0	0	289.0	3	46798	300.1	2	48541	3.72%
12	31	0	0	754.2	5	55610	900.0	2	57817	3.97%
15	36	0	0	1800.0	6	64478	1800.0	5	67486	4.67%
18	45	0	0	1800.0	24	80168	1800.0	16	84068	4.86%
20	51	0	0	1800.0	28	90961	1800.0	39	95238	4.70%
25	61	0	0	1800.0	76	107247	1800.0	62	112276	4.69%
30	71	0	0	1800.0	135	124052	1800.0	82	128494	3.58%
35	86	7	23	3600.0	242	131218	3600.0	167	132218	0.76%
40	100	8	27	3600.0	354	132526	3600.0	270	133489	0.73%
Avg	-	-	-	-	-	-	-	-	-	**3.66 %**

5.2 Analysis of weight factor α

To answer (Q.2) and (Q.3), we perform one week analysis. From equation 4.1, we make use of weight factor α to evaluate the total cost paid by requesters to get their tasks accomplished at each planning horizon. Total cost for each task is addition of its subtasks processing cost and travelling cost if job has travelled for subsequent subtasks from machine-to-machine as per schedule. In the set of experiments we evaluate three special cases, i) $\alpha = 1$, ii) $\alpha = 0$, iii) $\alpha = 0.5$. By setting $\alpha = 1$, objective function makes a decision on cost factor and try to minimize the overall cost of tasks. In setting $\alpha = 0$, scheduling decision is made on trust factor where objective function try to schedule maximum number of subtasks to trustworthy providers and $\alpha = 0.5$ prioritize both the factors equally.

To determine the significance, one week analysis is performed. For each horizon of one day, we vary amount of machine availability in time $a_k = [0,24]$, number of time windows $e = [1,3]$ and available time of windows $[w^e_{k_l}, w^e_{k_u}]$. Each day providers have an opportunity to set new hourly rate for providing machine that vary between $r_k = [500,750]$. Number of tasks and its subtasks are varied for each horizon. To replicate the practicality at BIC, we develop two situations. In situation one, machine requirements are less than available machines (i.e. Day 1-5) and in situation two machine requirements are more than available machines (i.e. Day 6-7). Thus, in total we evaluate 7 days scenario and the results are tabulated in Table 5.3. Once all the tasks are completed by the end of each horizon, requesters ratings are randomly created on the scale of [0,10] for each subtask. Ratings are recorded and updated with trust value update mechanism as explained in section 4.2. By the end of each horizon, trust value of each company is updated and tabulated in column 'Trust', Table 5.4. Number of scheduled subtasks are updated in column 'Allot' of Table 5.4. We start experimentation by giving equal trust value as 10 to each company, whereas scheduled subtasks are set to zero. To schedule for current horizon, we make use of company's updated trust value of previous horizon. Results tabulated in table 5.4 are entrenched with case (iii).

We have formulated a code in python that automatically generates a gantt chart after scheduling is completed. We fetch required output data through variables from optimization model which is given as input to generate chart. This helps to visualize the schedules of subtasks on providers machine. Gantt chart of Day-2 with $\alpha = 0.5$ can be visualized in Figure 5.6.

For each scenario, we record total amount of cost paid by requesters for overall tasks as well as travel by end of the day. Outputs are recorded for all cases (i), (ii) and (iii). Travel time is translated to travel cost. Gap 1, Gap 2 and Gap 3 are calculated as formulated in equation (5.1), (5.2) and (5.3). Gap 1 show that on average requesters have to pay 3 % extra cost when average gap between case (i) and (iii) is considered. Gap 2 indicates that on average requesters have to pay 9 % extra cost when average gap between case (ii) and (iii) is considered. Gap 3 indicates that on average requesters have to pay 12 % extra cost when average gap between case (i) and (ii) is considered. After analysing results we can say that by prioritizing cost factor in optimization model requesters have to pay lowest cost. While prioritizing trust factor, requesters may have to pay huge amount of cost. Giving equal weightage to both cost and trust factor come with some extra cost which is not huge but is always better than prioritizing trust factor. This answers our (Q.2).

$$\text{Gap 1} = \frac{\text{Equal weight(Processing cost + Travelling cost)} - \text{Cost factor(Processing cost + Travelling cost)}}{\text{Cost factor(Processing cost + Travelling cost)}}\% \tag{5.1}$$

$$\text{Gap 2} = \frac{\text{Trust factor(Processing cost + Travelling cost)} - \text{Equal weight(Processing cost + Travelling cost)}}{\text{Equal weight(Processing cost + Travelling cost)}}\% \tag{5.2}$$

$$\text{Gap 3} = \frac{\text{Trust factor(Processing cost + Travelling cost)} - \text{Cost factor(Processing cost + Travelling cost)}}{\text{Cost factor(Processing cost + Travelling cost)}}\% \tag{5.3}$$

Bar chart in Figure 5.3 exhibit total amount requesters has to pay after 7 days of sharing for getting their tasks processed and travel cost. Orange bar denote overall travel cost and blue bar denote process cost. Owing to companies situated in close vicinity, task processing cost are extremely high and always dominates travelling cost. To reduce overall cost of task it could be beneficial even if job in a task has to travel to multiple providers for processing multiple subtasks.

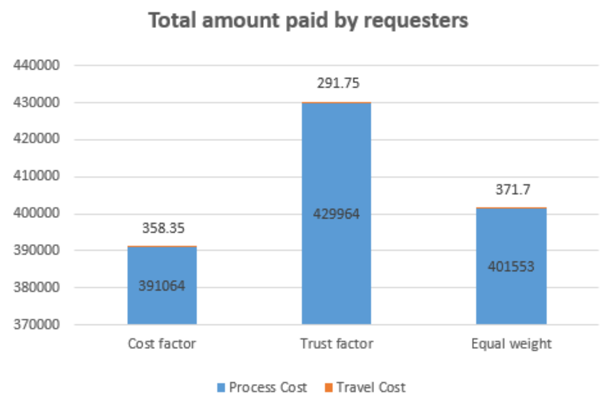

Figure 5.3: Total cost paid by requesters after 7 days of sharing

To see how beneficial resource sharing can be for providers. We record overall machine utilization rate of each company by equation (5.4), before sharing. After scheduling is done we fetch total number of subtasks scheduled to machines of each provider at free time slots and again calculate utilization rate of each company by equation (5.5), after sharing. In this way, utilization rate before and after sharing is documented for each horizon. In Figure 5.4 blue bar indicate 7 days average of utilization rate of each company before sharing, whereas orange bar indicate 7 days average of utilization rate of each company after sharing. On average each company gained some profit through sharing as their resource utilization has increased. If companies are not sharing machines, then companies with idle times are just useless and unproductive. Therefore, we consider that amount of price paid by requesters to providers is profit to them for getting task done. Profit gained by each company as provider is calculated with an equation (5.6) and profit gained can be seen in Figure 5.5. If we look at Figure 5.5, profit gained by company 5 and 6 is less which is justified because these two companies are small scale with less number of machines. However, company 2 being large scale company gained less profit. To scrutinize the reason if we look at the trust value in table 5.4, company 2 scored low trust value during horizon 2 and still it was pitching high hourly machine rate. Due to less number of subtasks from horizon 2-4 and plenty of scheduling options at other providers, company-2 did not receive any subtasks for three consecutive horizons. It was only considered when overall subtasks increased and available machines were less in horizon 6 and 7.

$$\text{Utilization before sharing} = \frac{\text{Time windows occupied by company to perform its own task}}{\text{Overall machine uptime}} \times 100 \tag{5.4}$$

$$\text{Utilization after sharing} = \frac{\text{Time windows occupied by company to perform its own task + Free time windows scheduled to requesters}}{\text{Overall machine uptime}} \times 100 \tag{5.5}$$

$$\text{Profit gain} = r_k \times \text{scheduled subtasks } \times t_{ij} \tag{5.6}$$

Resource sharing is profitable for companies from both sides. Company being requesters can get help from providers and can satisfy customer demands. Companies being providers can gain some extra profit by helping requesters with idle resource. It's a win-win situation for companies. This answers our (Q.3).

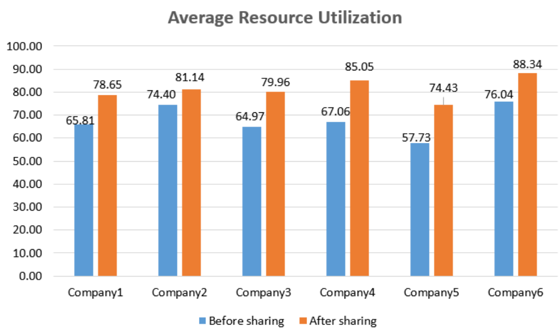

Figure 5.4: Average machine utilization rate

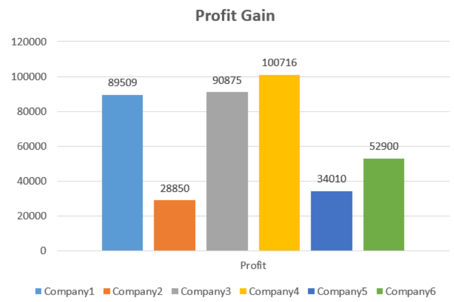

Figure 5.5: Overall profit gained by providers

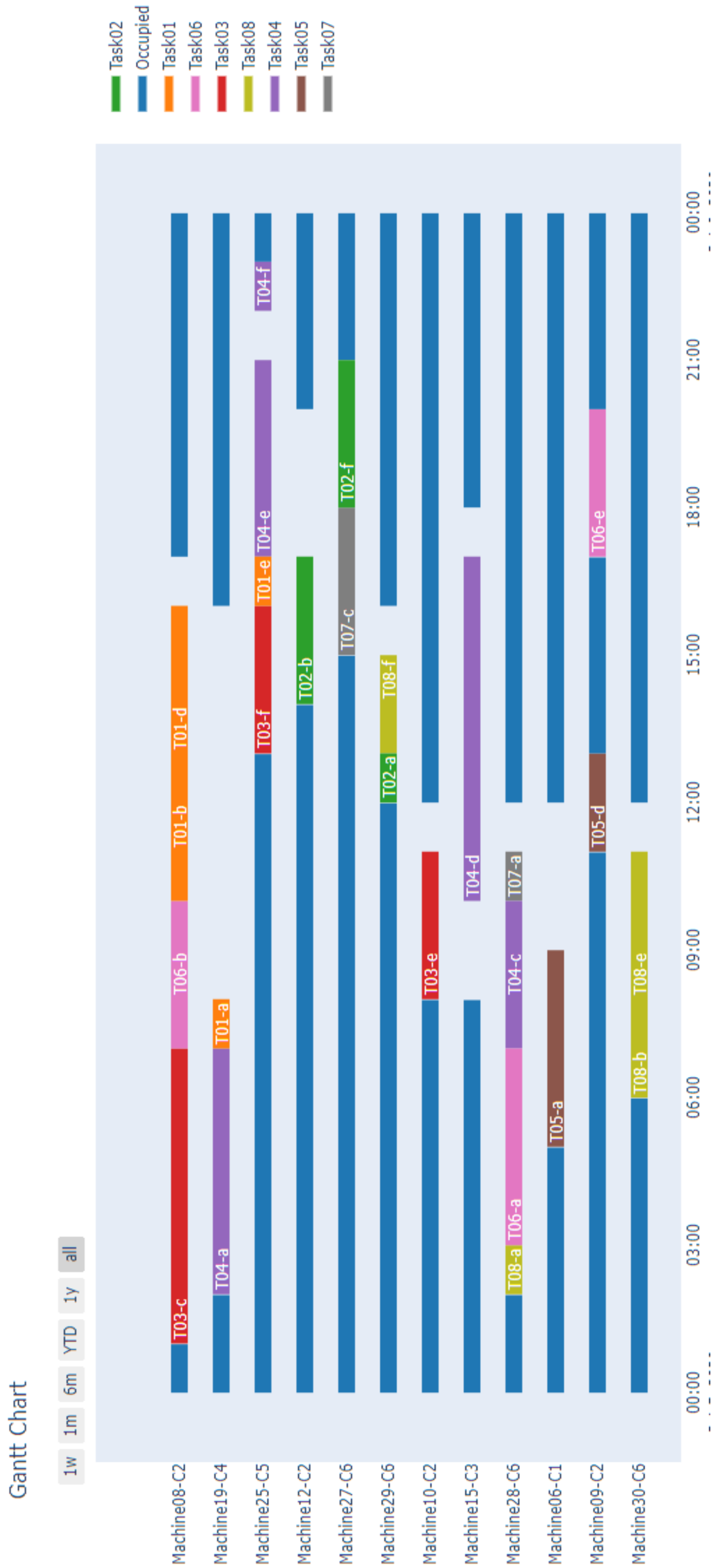

Figure 5.6: Scheduling results for day 2 with equal weightage of cost and trust factor i.e. $\alpha = 0.5$

Table 5.3: Results showing the effect of setting the weight parameter, α = 1, 0, 0.5 in optimization model

					Optimization Model								
					$\alpha = 1$		$\alpha = 0$		$\alpha = 0.5$				
Horizon (H)	**Tasks** (N)	**Subtask** (J)	**Cncl** (N)	**cncl** (J)	**Processing Cost**	**Travelling cost**	**Processing Cost**	**Travelling cost**	**Processing Cost**	**Travelling cost**	**Gap 1** (%)	**Gap 2** (%)	**Gap 3** (%)
Day1	4	12	0	0	9601	5.7	11535	21.9	9601	5.7	0%	20%	20%
Day2	8	26	0	0	42626	42	50019	33	46306	51.9	9%	8%	17%
Day3	10	29	0	0	47754	28.3	53752	16.5	48428	39	1%	11%	13%
Day4	12	35	0	0	56111	63	60232	41.1	56578	53.4	1%	6%	7%
Day5	14	44	0	0	65921	71.4	72600	42.3	67382	64.8	2%	8%	10%
Day6	16	50	2	7	88877	71.4	98534	65.2	89284	70.5	0%	10%	11%
Day7	20	65	2	6	80174	76.5	83292	71.7	83974	86.4	5%	-1%	4%
Average	-	-	-	-	-	-	-	-	-	-	**3%**	**9%**	**12%**

Table 5.4: Updated trust values after each planning horizon

	Start		**Day 1 (H_1)**		**Day 2 (H_2)**		**Day 3 (H_3)**		**Day 4 (H_4)**		**Day 5 (H_5)**		**Day 6 (H_6)**		**Day 7 (H_7)**	
	Trust	Allot	Trust	Allot	Trust	Allot	Trust	Allot	Trust	Allot	Trust	Allot	Trust	Allot	Trust	Allot
Company1	10	0	6.3	3	6.0	4	5.4	12	4.4	19	4.0	22	4.5	31	4.8	44
Company2	10	0	10.0	0	4.8	8	4.8	8	4.8	8	4.6	14	5.0	18	5.1	22
Company3	10	0	6.8	5	6.0	6	5.8	13	5.2	20	5.1	32	5.2	41	5.3	55
Company4	10	0	7.3	4	6.5	6	6.0	21	5.5	35	5.6	52	5.4	67	5.2	83
Company5	10	0	10.0	0	6.0	4	5.7	6	5.4	8	4.8	11	4.7	15	5.0	21
Company6	10	0	10.0	0	5.8	10	5.8	10	4.7	15	4.9	18	5.2	22	4.8	27

5.3 Sensitivity Analysis

To answer (Q.4) and (Q.5), we perform sensitivity analysis. Set of tasks N = (3,5,8,10,12,15) with subtasks J = (11,18,29,34,39,48). We generate data in such a way that company 1-2 has low machine hourly rate and high trust value, company 3 has high trust value and high machine hourly rate, company 4 has low trust value and low machine hourly rate, company 5-6 has low trust value and high machine hourly rate. Number of available time windows and their available times are fixed at all the machines with 25 % machine availability. We generate three different scenarios to answer the question. In scenario one, we consider companies sharing platform has homogeneous machines, therefore we fix same set of capabilities to all the machines owned by each company. In scenario two, we consider companies sharing BIC platform has moderately heterogeneous machines, therefore we set the machine capabilities such that 60-70 % of capability is matched. In scenario 3, we consider companies sharing BIC platform are highly heterogeneous, therefore we set the machine capabilities such that 30-40 % capability is matched. Experiments are performed on all the three scenarios and results are tabulated in Table 5.5. 'ASU %' (after sharing utilization) and 'Profit' are calculated as shown in equation (5.5) and (5.6).

If we look at the results in Table 5.5. In scenario one, none of the tasks got cancelled. In scenario two, one task got cancelled when demand was huge with 15 tasks. In scenario 3, tasks started getting cancelled even when demand was not huge and machines were not hardly utilized. It would be more beneficial for BIC to invite companies to share platform whose resources are homogeneous or moderately heterogeneous. Companies sharing platform with highly heterogeneous resources will not benefit individual company as a requester or provider, since companies would not be able to help each other to cope with market fluctuations.

To show which company would earn maximum profit, we take average of 'Profit' for all the instances in all the three scenarios and divide it by number of machines owned by each company. It is important to look at the profit per machine because number of machines owned by each company are heterogeneous in number. In general, companies with more number of machines would earn more profit as they have more machines to share. If we look at Figure 5.8, company 1-2 earned maximum profit per machine in all the three scenarios as we kept fixed high trust value and low price for their machines. Company 5-6 earned least profit as we kept high price for their machines and fixed low trust value, whereas company 3 and 4 were in middle. After analysis we can say that, companies with high trust values and low resource price would be preferred more and companies with low trust values and high resource price would be least preferred in the environment.

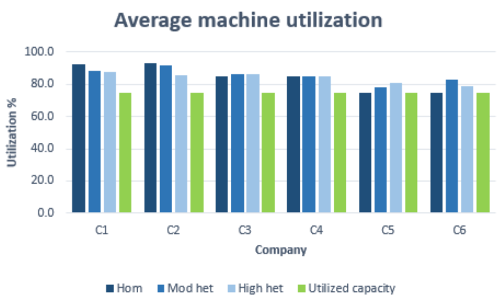

Figure 5.7: Average machine utilization

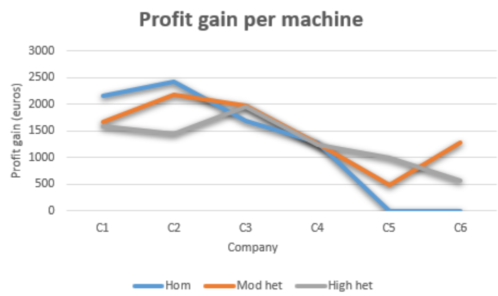

Figure 5.8: Profit gained by each company per machine

Table 5.5: Results of companies being homogeneous, moderately heterogeneous and highly heterogeneous with respect of machine

					Company1		**Company2**		**Company3**		**Company4**		**Company5**		**Company6**	
Scenario	N	J	cncl (N)	cncl (J)	Profit	ASU %	Profit	ASU %	Profit	ASU %	Profit	ASU %	Profit	ASU %	Profit	ASU %
hom	3	11	0	0	8569	85.1	5260	83.3	0	75.0	0	75.0	0	75.0	0	75.0
	5	18	0	0	13060	89.8	11564	92.5	2172	77.0	2016	77.7	0	75.0	0	75.0
	8	29	0	0	17304	94.6	12714	94.1	6528	81.8	7091	84.7	0	75.0	0	75.0
	10	34	0	0	16206	93.4	12710	94.1	14421	89.0	10515	89.7	0	75.0	0	75.0
	12	39	0	0	17318	94.6	14918	97.5	18666	93.0	12059	91.0	0	75.0	0	75.0
	15	48	0	0	18318	95.4	15438	98.2	19314	93.9	14569	96.4	0	75.0	0	75.0
	Avg				**15129**	**92.15**	**12101**	**93.3**	**10184**	**84.9**	**7708**	**84.9**	**0**	**75**	**0**	**75**
mod het	3	11	0	0	5144	80.9	6487	85.0	0	75.0	2529	78.4	0	75.0	0	75.0
	5	18	0	0	9962	86.3	7015	85.8	8024	82.6	3537	79.8	0	75.0	1956	78.1
	8	29	0	0	11026	87.5	13757	95.8	11605	86.0	6073	83.8	0	75.0	3350	80.2
	10	34	0	0	14238	91.0	12631	94.0	15988	90.2	8309	86.1	0	75.0	4047	81.3
	12	39	0	0	15841	92.9	12631	94.1	15988	90.1	9830	88.2	1977	81.2	8124	87.5
	15	48	1	3	13566	90.5	12664	94.2	19621	93.7	14987	94.4	3977	87.2	13159	94.8
	Avg				**11630**	**88.17**	**10864**	**91.5**	**11871**	**86.3**	**7544**	**85.1**	**992**	**78.1**	**5106**	**83**
high het	3	11	0	0	6333	82.1	2725	79.2	2147	77.0	3888	79.8	0	75.0	0	75.0
	5	18	0	0	8864	85.1	4270	81.6	11628	86.1	6561	83.3	0	75.0	0	75.0
	8	29	1	4	10918	87.5	8079	87.5	9535	84.0	9033	86.8	0	75.0	2160	78.1
	10	34	1	5	12014	88.7	8594	88.3	12419	86.8	6561	83.3	4658	89.5	4116	81.2
	12	39	2	6	12607	89.2	8079	87.5	16525	90.9	9035	86.8	4658	89.0	4166	81.2
	15	48	3	10	16153	93.4	11363	87.5	18127	92.3	9539	87.5	2696	83.0	3396	80.2
	Avg				**11148**	**87.67**	**7185**	**85.3**	**11730**	**86.2**	**7436**	**84.5**	**2002**	**81.1**	**2306**	**78**

Chapter 6

Extensions and Conclusion

6.1 Suggestion and extension

In this section we would like to give some suggestions for the practical implementation of framework to facilitate sharing and extension to developed model.

1. Framework proposed in section 3.4 automates the whole process of sharing. Since we do not consider any negotiation between requesters and providers. It is important that as a community, companies should decide a range of released hourly rate of machines that is acceptable by all companies as requesters. Between acceptable range, providers can release hourly rate of machine as per their trust value and amount of profit they want to gain. There can also be an alternative option where providers can pitch hourly rate to share its machine, but requester also release an upper limit they can pay their tasks on machines. We were not able integrate and test the alternative option due to limited time of research.
2. In this research, we made an assumption that if machine capability is matched for requested subtask, all capable machine at distinct providers take same amount of time to process it. But in practical situation, few companies might have old machines which require more time to process the same subtask than new machines at other companies. Therefore, future work may look at incorporating diverse processing time at different machines.
3. For each subtask of a task there would be some tool setup time before operation is performed and some cleaning time after operation is performed which can be considered as future research work to integrate into optimization model.

6.2 Future research

6.2.1 Full sharing

In this research, we explored the concept of partial sharing where each company shared resources owned by themselves. Having an opportunity to work on the same floor, companies at BIC can also together invest on expensive resources to reduce the financial burden. Resource sharing model can be developed which ensures that for each planned horizon how fairly resource can be allocated amongst companies as per their financial contribution to buy the resource. Extensively adapted fairness properties of game theory, multi-agent systems can be used to develop such model. Simulation study can be performed to extract insights on short term and long term benefits of full sharing.

6.2.2 Trust Mechanism

In this research, we have introduced trust update mechanism and shown its effect on scheduling results. For practical application it requires further research. There are different indicators on which requesters can rate providers. Depending on resource type, companies as a community needs to decide plausible indicators that should be considered for rating. In paper [Xu and Yu, 2014] different indicators are shown. Author has specified different methods where multiple indicators can be converted to a single rating value multiplying by weight factors. This can be further researched and integrated to trust value update mechanism developed in this paper to make trust concept practically applicable.

6.3 Conclusion

In this paper, we propose resource sharing model to schedule heterogeneous subtasks of tasks on heterogeneous set of machines. The heterogeneity arises at requesters side from differences in combination of subtasks and its processing time. The heterogeneity arises at providers side from differences in machine availability, time windows, cost and trust value. Proposed model can handle both the situation when overall tasks are more or less than available resources and generates best schedule. The main contribution of this paper are: i) Integrating multiple idle time windows of providers machines to schedule requesters tasks at heterogeneous set of machines ii) Introduced trust value update mechanism.

Looking at the experimental results we can say that resource sharing is beneficial for companies as a provider as well as requester. Although, integrating trust factor in decision making comes with 3-4 % extra cost. However, it ensures that most of the time requesters get trustworthy providers. Integrating trust factor also helps to maintain a competitive environment at shop floor. Since, companies would be aware that trust value is also considered in decision making of scheduling. Each company will try to maintain better trust value so that they get maximum tasks scheduled to their machine, hence maximize their profit. Provider that has maintained high trust value can release bit higher hourly rate and expects to get the tasks schedule on their machines. Whereas, provider that has lost its credibility in the environment need to release low hourly rate of machines to get the tasks until it regains its credibility. In future BIC might have multiple shop floors located at different locations in the campus, in that case travelling time and its cost would increase, however, this can be quite easily incorporated in the objective function.

Bibliography

[Becker and Stern, 2016] Becker, T. and Stern, H. (2016). Impact of resource sharing in manufacturing on logistical key figures. *Procedia CIRP*, 41:579 – 584. Research and Innovation in Manufacturing: Key Enabling Technologies for the Factories of the Future - Proceedings of the 48th CIRP Conference on Manufacturing Systems. 3, 6

[Benjaafar and Hu, 2020] Benjaafar, S. and Hu, M. (2020). Operations management in the age of the sharing economy: What is old and what is new? *Manufacturing & Service Operations Management*, 22(1):93–101. 3

[Benjaafar et al., 2019] Benjaafar, S., Kong, G., Li, X., and Courcoubetis, C. (2019). Peer-to-peer product sharing: Implications for ownership, usage, and social welfare in the sharing economy. *Management Science*, 65(2):477–493. 3, 4

[Cao et al., 2015] Cao, Y., Wang, S., Kang, L., and Gao, Y. (2015). A tqcs-based service selection and scheduling strategy in cloud manufacturing. *The International Journal of Advanced Manufacturing Technology*, 82. 5, 6, 8

[Chen et al., 2019] Chen, J., Huang, G. Q., Wang, J.-Q., and Yang, C. (2019). A cooperative approach to service booking and scheduling in cloud manufacturing. *European Journal of Operational Research*, 273(3):861 – 873. 1

[de Weerdt et al., 2012] de Weerdt, M., Zhang, Y., and Klos, T. (2012). Multiagent task allocation in social networks. *Autonomous Agents and Multi-Agent Systems*, 25(1):46–86. 4

[Elgendy et al., 2019] Elgendy, A., Yan, J., and Zhang, M. (2019). Integrated strategies to an improved genetic algorithm for allocating and scheduling multi-task in cloud manufacturing environment. *Procedia Manufacturing*, 39:1872 – 1879. 25th International Conference on Production Research Manufacturing Innovation: Cyber Physical Manufacturing August 9-14, 2019 | Chicago, Illinois (USA). 5, 6

[Freitag et al., 2015] Freitag, M., Becker, T., and Duffie, N. A. (2015). Dynamics of resource sharing in production networks. *CIRP Annals*, 64(1):435 – 438. 3, 4, 6

[Hu, 2019] Hu, M. (2019). *Sharing Economy: Making Supply Meet Demand*. Springer Series in Supply Chain Management. Springer International Publishing. 3

[Li et al., 2018] Li, K., Zhou, T., hai Liu, B., and Li, H. (2018). A multi-agent system for sharing distributed manufacturing resources. *Expert Systems with Applications*, 99:32 – 43. 5

[Liu et al., 2019] Liu, Y., Wang, L., Wang, X. V., Xu, X., and Zhang, L. (2019). Scheduling in cloud manufacturing: state-of-the-art and research challenges. *International Journal of Production Research*, 57(15-16):4854–4879. 2, 5, 6

[Ma et al., 2018] Ma, K., Wang, L., and Chen, Y. (2018). A resource sharing mechanism for sustainable production in the garment industry. *Sustainability*, 10(1):52. 3, 6

[Ramchurn et al., 2004] Ramchurn, S. D., Huynh, D., and Jennings, N. R. (2004). Trust in multi-agent systems. *The Knowledge Engineering Review*, 19(1):1–25. 5, 6

[Renna and Argoneto, 2011] Renna, P. and Argoneto, P. (2011). Capacity sharing in a network of independent factories: A cooperative game theory approach. *Robotics and Computer-Integrated Manufacturing*, 27(2):405 – 417. Translational Research â€“ Where Engineering Meets Medicine. 3, 4

[Sarker, 2002] Sarker, B. R. (2002). A review of: â€œsupply chain management: Strategy, planning and operationâ€ sunil chopra and peter meindl prentice hall, inc., 2001, 457 pages, isbn: 0-13-026465-2. *IIE Transactions*, 34(2):221–222. 1

[Xu and Yu, 2014] Xu, X. and Yu, H. (2014). A game theory approach to fair and efficient resource allocation in cloud computing. *Mathematical Problems in Engineering*, 2014:1–14. 4, 6, 28

[Ye et al., 2017] Ye, Q., Zhang, Y., and Dekker, R. (2017). Fair task allocation in transportation. *Omega*, 68:1 – 16. 4

[Ye and Zhang, 2016] Ye, Q. C. and Zhang, Y. (2016). Participation behavior and social welfare in repeated task allocations. In *2016 IEEE International Conference on Agents (ICA)*, pages 94–97. 4

[Zhou et al., 2018a] Zhou, L., Zhang, L., Sarker, B. R., Laili, Y., and Ren, L. (2018a). An event-triggered dynamic scheduling method for randomly arriving tasks in cloud manufacturing. *International Journal of Computer Integrated Manufacturing*, 31(3):318–333. 5, 6, 8, 9

[Zhou et al., 2018b] Zhou, L., Zhang, L., Zhao, C., Laili, Y., and Xu, L. (2018b). Diverse task scheduling for individualized requirements in cloud manufacturing. *Enterprise Information Systems*, 12(3):300–318. 5

[ÃdÃ¡m Szaller et al., 2020] ÃdÃ¡m Szaller, Egri, P., and KÃ¡dÃ¡r, B. (2020). Trust-based resource sharing mechanism in distributed manufacturing. *International Journal of Computer Integrated Manufacturing*, 33(1):1–21. 4

Appendix A

A.1 Travelling time Matrix

Matrix below represents travelling between machines in minutes.

	M01	M02	M03	M04	M05	M06	M07	M08	M09	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19	M20	M21	M22	M23	M24	M25	M26	M27	M28	M29	M30
M01	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5	8.0	8.5	9.0	9.5	10.5	11.0	11.5	12.0	12.5	13.0	13.5	14.0	14.5	15.0
M02	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5	8.0	8.5	9.0	9.5	10.5	11.0	11.5	12.0	12.5	13.0	13.5	14.0	14.5
M03	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5	8.0	8.5	9.0	9.5	10.5	11.0	11.5	12.0	12.5	13.0	13.5	14.0
M04	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5	8.0	8.5	9.0	9.5	10.5	11.0	11.5	12.0	12.5	13.0	13.5
M05	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5	8.0	8.5	9.0	9.5	10.5	11.0	11.5	12.0	12.5	13.0
M06	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5	8.0	8.5	9.0	9.5	10.5	11.0	11.5	12.0	12.5
M07	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5	8.0	8.5	9.0	9.5	10.5	11.0	11.5	12.0
M08	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5	8.0	8.5	9.0	9.5	10.5	11.0	11.5
M09	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5	8.0	8.5	9.0	9.5	10.5	11.0
M10	4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5	8.0	8.5	9.0	9.5	10.5
M11	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5	8.0	8.5	9.0	9.5
M12	5.5	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5	8.0	8.5	9.0
M13	6.0	5.5	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5	8.0	8.5
M14	6.5	6.0	5.5	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5	8.0
M15	7.0	6.5	6.0	5.5	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5
M16	7.5	7.0	6.5	6.0	5.5	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0
M17	8.0	7.5	7.0	6.5	6.0	5.5	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5
M18	8.5	8.0	7.5	7.0	6.5	6.0	5.5	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0
M19	9.0	8.5	8.0	7.5	7.0	6.5	6.0	5.5	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5
M20	9.5	9.0	8.5	8.0	7.5	7.0	6.5	6.0	5.5	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
M21	10.5	9.5	9.0	8.5	8.0	7.5	7.0	6.5	6.0	5.5	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5
M22	11.0	10.5	9.5	9.0	8.5	8.0	7.5	7.0	6.5	6.0	5.5	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0
M23	11.5	11.0	10.5	9.5	9.0	8.5	8.0	7.5	7.0	6.5	6.0	5.5	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5
M24	12.0	11.5	11.0	10.5	9.5	9.0	8.5	8.0	7.5	7.0	6.5	6.0	5.5	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5	3.0
M25	12.5	12.0	11.5	11.0	10.5	9.5	9.0	8.5	8.0	7.5	7.0	6.5	6.0	5.5	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0	2.5
M26	13.0	12.5	12.0	11.5	11.0	10.5	9.5	9.0	8.5	8.0	7.5	7.0	6.5	6.0	5.5	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5	2.0
M27	13.5	13.0	12.5	12.0	11.5	11.0	10.5	9.5	9.0	8.5	8.0	7.5	7.0	6.5	6.0	5.5	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0	1.5
M28	14.0	13.5	13.0	12.5	12.0	11.5	11.0	10.5	9.5	9.0	8.5	8.0	7.5	7.0	6.5	6.0	5.5	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5	1.0
M29	14.5	14.0	13.5	13.0	12.5	12.0	11.5	11.0	10.5	9.5	9.0	8.5	8.0	7.5	7.0	6.5	6.0	5.5	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0	0.5
M30	15.0	14.5	14.0	13.5	13.0	12.5	12.0	11.5	11.0	10.5	9.5	9.0	8.5	8.0	7.5	7.0	6.5	6.0	5.5	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	0

Figure A.1: Travel time

A.2 Machine Capabilities

Number of machines owned by each company, it's machine id and capabilities are documented in table A.1 below.

Table A.1: Capabilities of machines

Machine_id	Company	Homogeneous capability	Moderately heterogeneous capability	Highly heterogeneous capability
Machine01.0	1	["a","b","c","d","e","f"]	["a","b","c"]	["a","b"]
Machine02.0	1	["a","b","c","d","e","f"]	["c","e","f"]	["a","b"]
Machine03.0	1	["a","b","c","d","e","f"]	["c","d","e"]	["c","e","f"]
Machine04.0	1	["a","b","c","d","e","f"]	["a","b","c","f"]	["c"]
Machine05.0	1	["a","b","c","d","e","f"]	["c","d","e"]	["c","f"]
Machine06.0	1	["a","b","c","d","e","f"]	["a","b"]	["f","d"]
Machine07.0	1	["a","b","c","d","e","f"]	["c","e","f"]	["a","e"]
Machine08.0	2	["a","b","c","d","e","f"]	["d","b","c"]	["e"]
Machine09.0	2	["a","b","c","d","e","f"]	["c","d","e"]	["d","c"]
Machine10.0	2	["a","b","c","d","e","f"]	["c","d","e"]	["c","d","e"]
Machine11.0	2	["a","b","c","d","e","f"]	["c","e","f"]	["c","d","e"]
Machine12.0	2	["a","b","c","d","e","f"]	["b","c"]	["e"]
Machine13.0	3	["a","b","c","d","e","f"]	["a","d","e"]	["c","e","f"]
Machine14.0	3	["a","b","c","d","e","f"]	["b","d","e"]	["b","c"]
Machine15.0	3	["a","b","c","d","e","f"]	["c","e","f","d"]	["a","d"]
Machine16.0	3	["a","b","c","d","e","f"]	["c","d","e"]	["b","d","e"]
Machine17.0	3	["a","b","c","d","e","f"]	["a","b","c"]	["b","d"]
Machine18.0	3	["a","b","c","d","e","f"]	["a","e","f"]	["a"]
Machine19.0	4	["a","b","c","d","e","f"]	["a","b","c","e","f"]	["a"]
Machine20.0	4	["a","b","c","d","e","f"]	["a","b","c"]	["f"]
Machine21.0	4	["a","b","c","d","e","f"]	["c","e","f"]	["a","b","c","e","f"]
Machine22.0	4	["a","b","c","d","e","f"]	["a","b","c"]	["d"]
Machine23.0	4	["a","b","c","d","e","f"]	["c","d","e"]	["c","e","f"]
Machine24.0	4	["a","b","c","d","e","f"]	["a","b","c"]	["a"]
Machine25.0	5	["a","b","c","d","e","f"]	["e","f"]	["a"]
Machine26.0	5	["a","b","c","d","e","f"]	["c","d","e"]	["e"]
Machine27.0	6	["a","b","c","d","e","f"]	["c","e","f"]	["a","b","c"]
Machine28.0	6	["a","b","c","d","e","f"]	["a","c"]	["e","f"]
Machine29.0	6	["a","b","c","d","e","f"]	["a","e","f"]	["c","d","e"]
Machine30.0	6	["a","b","c","d","e","f"]	["b","d","e"]	["f"]