Task Allocation

A Game theoretical approach in Project Management

Arpit Gupta **1207301364**
Mukund Manikarnike **1208597425**
Nitish Bhatia **1207300584**
Surbhi Aggarwal **1208920371**

Game Theory with Applications
CSE 556 - Project
Fall 2015

school of computing, informatics, & decision systems engineering

ASU IRA A. FULTON SCHOOLS OF engineering

Contents

List of Figures

Figures

1. Abstract

This paper focuses on application of game theory in software development project management strategies. The most important part of a successful project is a competent Task management strategy. It includes distributing the tasks in a project among the team members (Software engineers, Quality testers, Project planners etc.) in a way that will motivate every team member to give their best efforts which in turn will aid in rewarding outcome as the project finishes. This paper takes the ideas from the auction concepts and distributes the tasks to the team members by calculating the best allocation strategies considering various variables like time to complete tasks, preference factors, effort estimates. The project manager can use this strategy to compute the utilities for various stages in a project and distribute the task in a way that will satisfy the team members as well as yield a successful final product.

Keywords

Software Project Management, Game theory, Task Assignment, Auction, effort estimate, Project Manager, Team Members

2. Introduction

2.1. Problem Overview

A project is a well-constructed approach to problem solving consisting of several activities and operations accomplished in a certain order (or simultaneously) to reach a certain predefined goal[4]. Every Software project has a unique and affirmative objective with a start and end time and is usually months to years long. It is an exhaustive process of information and requirement gathering to coding to testing and maintenance[4].

Tasks can vary from each other, differentiated by complexity, from low to high[4]. If we consider Software development, tasks depends on multiple stages and deliverables like project analysis, data collection, study, system design, coding, database design, program design and coding, testing (Unit and System testing) and many more depending on the Software projects.

Effective task assignment will involve administering all aspects of a task, including the status of the task, priority at which task must be conducted, ideal time taken for a task to complete and expected time taken, cost allocation, human resource allocation.

2.2. Goal Description

We focus on performing an effective task assignment using a game theoretical approach so that the following factors are taken into account.

1. The team members are all satisfied with the tasks assigned to them.
2. The overall performance of the given tasks to the available users maximizes the total gain of the organization
3. There is an equilibrium in the task assignment so that the team members don't switch companies.

2.3. Organization of the Paper

A brief description of the document organization is given below.

1. **Section 3** talks about the methodologies followed including how the game has been formulated, the utilities of the each player in the game and how task assignment is carried out.
2. **Section** 4 talks about how the model that we propose plays out theoretically and how it compares with existing studies.
3. **Section 5** talks about the approaches we used before solving this model and what real-world problem this model solves and how it does the same.

3. Methodology

3.1. Terminologies

The problem is played out as a q-stage game. This section describes the players involved in the game and their attributes.

3.1.1. Players

3.1.1.1. Project Manager

A project manager is a professional in the software project who is responsible for the planning, execution and the completion of the project in a specified interval of time, typically relating to construction industry, architecture, aerospace, defense, but we will be specifically talking about Computer Software projects. The project manager handles the development of the project, most importantly the task assignment among his team members. He makes sure that the project is completed in time and under the predefined cost constraints.

3.1.1.2. Team Members

For a particular project, every project manager has a set of team members working under his supervision. The project manager assigns tasks to all the team members based on their capabilities, and their choice. The objective of this paper is to make a game model which will help the project manager to assign the tasks efficiently and to extract highest potential from its team members.

3.1.2. Attributes

The project manager has to allocate a set of n tasks $T_i = \{ T_1, T_2, ... , T_n \}$ in a Software Project among m team members, $U_i = \{U_1, U_2, U_3, ... , U_m\}$ where the number of tasks are greater than and equal to the number of team members. The team members are the software developers, project planning experts, and testers who work under the supervision of the Project Manager and are interested in working for the Project P. We assume here that the number of tasks are always greater than or equal to the number of team members.

Each team member has a preference factor associated with him, $pi = \{p_1, p_2, p_3, ... , p_m\}$.

Assignment is concluded of the tasks among the team members and either one or more than one task is assigned to a particular team member. There are n tasks in the project and the assignments or tasks are represented as $a_i = \{a_1, a_2, a_3, ... , a_n\}$.

Preference of a team member represents the order in which he wants the project tasks. It means that each team member will have a set of preferences in which he wants to do the tasks. For example, if there are three tasks, T_1, T_2, T_3 and the team member U_1's set of preferences for the three tasks are - T_2, T_3, T_1. It means that the team member U_1's satisfaction factor will be calculated as follows if he gets assigned T_3.

$$\frac{a_i}{p_i} = \frac{2}{3}$$

This is because he was assigned his 2^{nd} best preference out of the 3 tasks available.

Each task is associated with a time span in which it should be ideally completed. We consider the time taken by the team member to complete a particular task and it is denoted as follows, $t_i = \{t_1, t_2, t_3, ... , t_n \}$.

The tasks are also associated with effort estimate, the amount of effort it would take a particular team member to complete the task in a specified time. The effort estimate is represented as $e_i = \{ e_1, e_2, e_3, ... , e_n\}$.

3.2. Game Model

In order to describe the game model, we first start with a few assumptions where the task assignment is carried out in project management in the form of a game and then we go on to describe the algorithm that the game employs to achieve its purpose.

3.2.1. Assumptions

We assume that the game plays out under the following assumptions

1. Every team member will be assigned at least one task.

2. Every member exhibits a certain amount of interest for all tasks.
3. Every member has full information about all the tasks available.
4. Every task would be assigned to neither greater nor lesser than 1 team member.

3.2.2. Game Scenario

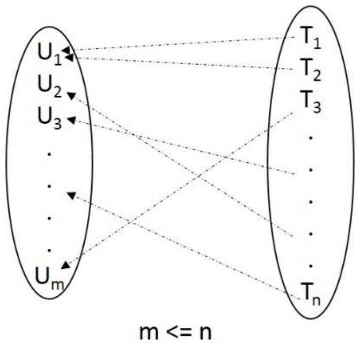

Figure 1 Game Scenario

The game consists of m number of users and n number of tasks with an assumption that the number of tasks are always greater than or equal to the number of users (n >= m). As seen in **Figure 1**, the tasks are shown on the right side of the graph and the users or team members are shown on the left side. Each team member can be assigned one or more than one task (as n >= m).

3.2.3. Game Outline

Every project can be divided into a set of sprints where a certain set of tasks are carried out in each sprint. The game model that we describe takes into account the following two factors in the utility of a team member

1. Satisfaction factor of each team member
   - This attribute gives us an idea about how satisfied each member is with the task that is assigned to him.
2. Performance index of each team member in the previous stage.
   - This attribute gives us an idea of how well a member performed in the previous stage. By using this, we ensure that there is a feedback mechanism being taken forward from stage to stage.

The above 2 factors put together also decide how good a task assignment is, for both the team members and the team in general. Certain quantified details are provided as part of **3.3**.

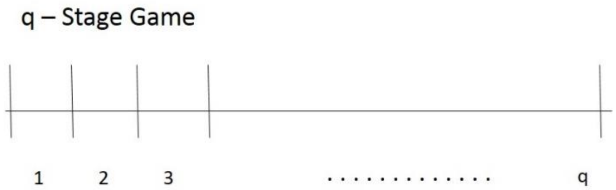

Figure 2 q-Stage Game

The q-stage game illustrated in **Figure 2** which considers each sprint as a stage, is a finitely repeated game, each stage of which proceeds as follows.

1. For each such stage, we compute the utilities of each of the team members for each of the task allocations are computed. The task allocation function described in **3.3** takes care of performing the best assignment.
2. However, the first stage of the game will only consider the satisfaction factor of each team member because we do not have any information about the performance of a player in a previous stage.

3.3. Utilities & Task Allocation

As described in **3.2.3**, every team member is associated with a preference $p_{i,j}^{q}$ and a corresponding assignment of tasks $a_{i,j}^{q}$. These team members also estimate the efforts for each task $e_{i,j}^{q-1}$ and complete their tasks in a certain time $t_{i,j}^{q-1}$. The utility at q^th stage is calculated as follows.

$$u_{i,j}^{q} = \left(\frac{a_{i,j}^{q}}{p_{i,j}^{q}} \times \frac{t_{i,j}^{q-1}}{e_{i,j}^{q-1}}\right)$$

$u_{i,j}^{q} \rightarrow utility\ of\ user\ i\ at\ q^{th}\ stage\ for\ j^{th}\ task.$

$$u_i^q = \sum_{j=1}^{n}\left(\frac{a_{i,j}^q}{p_{i,j}^q} \times \frac{t_{i,j}^{q-1}}{e_{i,j}^{q-1}}\right)$$

$$u_i^q \rightarrow utility\ of\ user\ i\ at\ q^{th}\ stage$$

$$(q-1) \rightarrow represents\ the\ previous\ stage$$

In the above utility functions, the first ratio in the product is referred to as the team member satisfaction factor and the second ratio is referred to as the performance index of the team member.

These utility functions work well for any stage, q > 1. However, if we were to look at the first stage, then we wouldn't have any prior information about the performance of the team member. Hence, we assume that the performance index is one. Hence, the utility function stage 1 is given as shown below.

$$u_i^q = \sum_{j=1}^{n}\left(\frac{a_{i,j}^q}{p_{i,j}^q} \times 1\right) = \sum_{j=1}^{n}\left(\frac{a_{i,j}^q}{p_{i,j}^q}\right)$$

As described in **3.2.3**, we know that there are n tasks and m users. Hence, we could allocate tasks in P_m^n number of ways, i.e. number of permutations when there are n tasks and m team members. In order to find the best allocation possible, we define what is called task allocation function as shown below.

$$\tau_x^q = \sum_{i=1}^{m}\sum_{j=1}^{n} u_{i,j}^q \times e_{i,j}^q$$

$$\tau_x^q \rightarrow Task\ allocation\ fn.for\ the$$

$$q^{th} stage\ for\ x^{th} allocation\ vector$$

The task allocation function is computed for each such task allocation vector and the task allocation that produces the lowest possible result is chosen as the best assignment.

4. Evaluation

4.1. Theoretical Analysis

This section offers a theoretical analysis of the model proposed for the problem that we aim to solve. As described in each of the sections that follow, the model that we proposed satisfies the desirable properties in any game theoretical approach.

4.1.1. Truthful

The task allocation game theoretic model is truthful as no team member has any incentive to report wrong estimate of the efforts required to complete the task. The argument for the same is that the task allocation used in every sprint to compute the task allocation to team members considers both the estimate of the efforts required to complete the task as well as the performance index of each team member from the previous stage.

$$\tau_x^q = \sum_{i=1}^{m}\sum_{j=1}^{n} u_{i,j}^q \times e_{i,j}^q$$

$$\tau_x^q \rightarrow Task\ allocation\ fn.for\ the$$

$$q^{th} stage\ for\ x^{th} allocation\ vector$$

If a member reports the false estimate of the effort, he may end up not getting the preferred task because if the reported estimate is very less than the actual time taken by him to complete it, will eventually affect his performance index which will impact his task assignment in later stages. Moreover, if the reported estimate is more than the actual time taken by him to complete the task, will affect his assignment for the current stage itself as he may not end up getting that task and some other member who might report the truthful estimate may end up winning the task. This proves that every team member must report truthful estimate of their efforts required to complete the task to maintain their performance index and stand in the race of task allocation.

4.1.2. Individual Rationality

The task allocation game theoretic model is individual rational as every team member in each sprint for the allocated task has positive utility.

$$u_{i,j}^{q} = \left(\frac{a_{i,j}^{q}}{p_{i,j}^{q}} \times \frac{t_{i,j}^{q-1}}{e_{i,j}^{q-1}} \right)$$

$u_{i,j}^{q} \rightarrow utility\ of\ user\ i\ at\ q^{th}\ stage\ for\ j^{th}\ task.$

The utility function is the multiplication of satisfaction ratio and performance index. Since, in every sprint each member provides his preference order of tasks and each member is assigned at least one task in each sprint employs that the satisfaction ratio will always be positive. Moreover, performance index will also be positive always as it is a measure of the variance of the reported estimate and the actual time taken in the previous sprint. Performance index is either in the range (0 1] or [1 ∞).

4.1.3. Member Sovereignty

The task allocation game theoretic model takes care of the assignment in a way that every individual member is assigned some task in each sprint and that every task is assigned to at least one individual in each sprint. This is ensured by the task allocation function which computes the task allocation factors for allocation vectors that satisfy this constraint.

4.1.4. Computationally Efficient

The task allocation game theoretic model is computationally efficient as the utility function and the task allocation function can be computed in deterministic time. Each project has a number of phases in which it is completed, and each phase has a project manager associated with it with a number of team members working under the project manager's supervision. This game theoretical approach can be used by each project manager to assign the tasks to his team members. So, the numbers which we are dealing in our utility functions are not enormous and will frequently be less than 20.

4.1.5. Nash Equilibrium

The task allocation game theoretic model will always exhibit a Nash Equilibria. Nash Equilibria in this model exists when each member's estimation in each sprint is optimal given the estimation of other members. In other words, no member would choose a different estimation as long as the estimation of other members remains the same.

The Game theoretic model proposed in this paper always achieve optimal allocation of tasks as in each sprint no member has any incentive to report untruthful estimate of the efforts required to complete the task. Every member reports his correct estimation in order to maintain a good performance index which affects the task allocation. The task allocation function is designed in such a way that it not only takes into account the reported estimation of efforts by members but also their performance index computed from the previous sprints. Hence, no player gets an incentive to report wrong estimations which makes the model truthful. In other words, true estimations of efforts from members and use of the performance index of individual member from the previous sprints, always results in an optimal allocation of tasks. Since, it is a finite staged game, therefore optimal allocation of tasks which is the Nash Equilibria is employed in each sprint.

4.2. Comparative Study

We look at other approaches that have been used as part of this section.

1. We saw studies involving the time, money and features required to be developed for a project being used as a factor to maximize the utilities of both, the employees as well as the organization [1].
2. We saw other studies which used game theory in task assignment which provided certain game theoretic algorithms to solve the problem at hand[2]

But, none of these models are modelled close to a real world scenario like the model we proposed takes into account, as described in **Section 5**. By using our model, the existence of a Nash equilibrium also has been analyzed which is of paramount importance because that would give us an idea of whether our model would survive in the real world.

5. Conclusion

5.1. Roadmap

This section talks about the ideas that were analyzed before the game model discussed in this paper was used as a solution to the problem that we describe.

The problems we discussed were

1. The problem of finding an equilibrium in a game where different employees think of different individual returns from a project whereas the goal is to maximize the utility or return from the project using references[1].
2. We also thought of modelling a single stage game where the utilities of employees and their preferences need to be taken into account to find an equilibrium

However, we chose the game that we described in this paper because an n-stage game is a lot closer to the real-world scenario. Additional details are described in **5.2**.

5.2. Summary

We have analyzed the subject of task assignment in project management by modelling it as an n-stage game. The strong points that we would like to highlight are

1. An n-stage game is a closest to real-world scenarios.
2. The utility function provides a feedback mechanism from each stage to the next one. Although, each stage ensures only close to fair assignment, since we use the feedback mechanism, fairness in the n-stage mechanism for averages out and tends to an equilibrium which is the desired result from this task.
3. The solution proposed here factor in the user satisfaction. The main contribution in this area that we provide from this paper is that we quantify the satisfaction that each employee gets. By quantifying something as abstract as this, as a project manager, we can ensure that employee retention is high.

5.3. Future Work

The biggest assumption that we make as part of this paper is that teams are broken down into sufficiently small teams with lesser team strengths. Hence, the model works out to be computationally efficient. If this model were to be applied to any team with a larger team strength, it would take lots of time to compute each of the task allocation functions for each of the task allocation vectors. Further refinement to this approach to treat such a scenario would be one area of work that we could think of for this model.

6. Bibliography

[1] Anurag Shrama, SDET at Intel Security *https://www.linkedin.com/pulse/applying-game-theory-software-project-estimation-anurag-sharma*

[2] Brent Lagesse, University of Texas at Arlington, Department of Computer Science and Engineering *Game-Theoretical Model for Task Assignment in Project Management* Management of Innovation and Technology, 2006 IEEE International Conference

[3] Software Project Management - Wikipedia *https://en.wikipedia.org/wiki/Software_project_management*

[4] Task Management – Wikipedia *https://en.wikipedia.org/wiki/Task_management*