Abstract

While there have been some efforts to compare centralized versus market based approaches to general task allocation problems, no effort has been made to compare those two approaches to mobile task allocation problem which can be characterised with uncertainty and high frequency of exceptions during the execution of generated schedules. This paper aims to fill the gap by using computer simulation in which uncertainty and exceptions are introduced in mobile working environment and the generated schedules from those two approaches are compared to find conditions that make one approach superior to the other.

1. Introduction:

Task allocation problem (TAP) involves the assignment of a number of workers (resources) to meet the requirements of a number of tasks (consumers). In usual cases the total demand exceeds the resources available, the challenge is to find the best combination of resource-consumer pairs in a way that will improve the performance of the resource provider (organization) according to a predefined measure i.e. the total profit of the organization (Cheng, Wellman, and Perry 2003).

Motivated by this point, many approaches to solve TAPs appeared according to differences in the type (machinery, human, etc…) and status (centralized, fixed, static,
etc...) of resources, arrival of tasks (deterministic, constant, etc...) to serve, and the task environment in general (dynamic or static).

The proposed approaches can be clustered in two main categories: centralized and decentralized approach. Centralized approach relies on a well-informed central decision point to produce task allocations and is widely used in industrial engineering like scheduling parallel machines in a factory (Cheung and Zhou 2001). Techniques like operational research (OR) and heuristic algorithms are the most used tools to solve TAP’s in a centralized way (Shen 2002). Centralized approach can produce optimal (or near optimal) allocation solutions, however it mainly deals with environments where global information is easily accessible.

In decentralized approaches, decisions are made by multiple decision makers, rather than a central one, based on the local knowledge and interactions of these decision makers who are the system members. Market-based mechanisms proved to be one of the most appropriate ways to implement decentralized approach because of its use of market principles to regulate members’ interactions. Market-based approach is applied for bandwidth allocation in telecommunications (Dramitinos et al. 2004) and robots coordination (Jones, Dias, and Stentz 2006). Market-based approaches have mainly been used more in environments where instant accurate global information is hard to gather.

The existence of both approaches opened the door to many researches to compare them trying to identify the key features of each one over the other in particular TAP, for example: Tan and Harker (1999) compared both approach in a workflow environment from a cost perspective, Ygge and Akkermans (1999) also did a comparison but from a performance point of view.
However these comparisons are applicable only for limited TAPs where resources and demand are mainly located in immobile environments, and don’t discuss the particular case of mobile task allocation problem (MTAP), besides, the criteria discussed largely differ from those of interest in MTAP.

Mobile Task Allocation Problem (MTAP) is a variation of the general TAP, where resources and tasks are distributed in a predefined area, and besides of being distributed, resources are mobile (physically moving as time progresses) and have to move (travel) to complete tasks according to its plan (schedule) facing environment’s exceptions which are highly increased because of resources’ mobility. The presence of travel and setups costs prior of fulfilling the tasks and the environment’s dynamism clearly identifies MTAP from the general TAP as schedules may endure great changes during execution caused by rescheduling as responses to environment uncertainty and exceptions handling.

Our main goal of this research is to compare both approaches for MTAP in a way where we can identify when and how to adopt an approach rather than the other. The comparison will be based on performance as the basic criterion.

We also aim to get a more concrete and global view of each approach in our particular MTAP in a way we can have a better understanding of each one, and according to the comparisons results, we’ll gain better knowledge to assist managers in dynamic and mobile businesses to decide which approach would perform better to manage business resources.

The research is conducted by analyzing numerical results obtained from an agent-based computerized simulator consisting of two major phases: planning phase in which schedules are generated, and execution phase during which schedules are executed in a dynamic mobile environment.
Simulation methods proved to be a good multi-purpose tool for researchers in social science, as Axelrod (1997) mentioned many advantages of simulation like usefulness to provide proofs, discovery of new principles and relationships, plus other advantages including extensibility, the ease of model understanding though complexity of expected results, results history tracking, and simplicity of data gathering, all among other important advantages.

The rest of this paper will include a description of the target problem in section 2. The simulation models adopted in this paper will be explained in section 3. Some initial results will be listed in section 4, and finally the foreseen future work is added in section 5.

2. Problem description

We consider a mobile working environment for the comparison of the two approaches in the simulation. Tasks (consumers) that need to be completed are distributed over geographical area and mobile workers (resources) move from one task location to others based on their schedules. Each worker has different starting and finishing time for each different working day; that is the availability of workers is limited. Third, each worker has a starting location each day. The scheduling process should consider travelling from each worker’s starting location to the first scheduled task location.

To differentiate between tasks’ priorities, we couple each task with a bonus score reflecting its importance. Workers aim to maximize their collected bonus points by completing tasks.

The global objective is to maximize total bonus points collected by workers with respect to scheduling constraints (like schedule lengths, tasks deadlines, etc…), it can be mathematically represented:
\[
\max G = \sum_{w=1}^{N} U_w \quad \text{Where for each worker } w:
\]

\[
U_w = \sum_{x=1}^{n} u_x \quad \text{And} \quad \sum_{x=1}^{n} D_x \leq \text{ScheduleTo talTime}
\]

\( G \): is the global optimization factor to maximize.
\( U_w \): Utility value gained by each worker.
\( N \): Number of workers.
\( u_x \): Utility gained from scheduling task \( x \).
\( n \): Number of scheduled tasks for a certain worker.

And \( D_x = \) duration of task \( x \) + travel time from task \( x-1 \) to task \( x \). When \( x-1=0 \) means the worker is travelling from his initial location to the first task.

Dynamism of the environment can be expressed in exceptions occurring while workers execute their schedules either during travel, setup (i.e. just before executing tasks), or during tasks’ execution. These exceptions range from simple delays to task or even schedule cancellation.

When an exception occurs, it must be handled in a way to minimize its impact. This can be done by rescheduling strategies. Rescheduling a task may be by assigning it to another worker.

3. Simulation Model

Our research model to drive the comparison will be by implementing an agent-based simulation model that will reflect our MTAP; we consider an environment consisting of 150 randomly-generated tasks to be completed by 25 workers located at random initial locations in a square geographical zone with a side-length value of 100. Each worker has a 7-hours schedule to travel and execute tasks. Travel distances are calculated according to the Euclidean metric.
\[ |L_1, L_2| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \]

Where: \( L_1(x_1,y_1) \) and \( L_2(x_2,y_2) \).

The simulation is divided into two phases: planning phase and execution phase. During the planning phase, schedules are statically created for workers. In the execution, these schedules are to be executed while facing environment dynamism. All exception handlings are performed during the execution phase.

3.1 Schedule generation

We use two algorithms to compare the approaches; the Hill-Climb algorithm is representing the centralized approach, and an economic variation of the Contract-Net Protocol (CNP) represents the market-based one.

**Hill-climb heuristic approach (Centralized)**

In centralized approach, schedules are created by a central point aware of all system values (tasks properties, workers initial location, etc…) and use hill climb heuristic to generate schedules.

We consider Hill-climb heuristic as it’s considered simple to implement and suitable for optimization problems.

Hill climb starts with a random initial solution, generally far from optimal, then iteratively improving the solution by looking in the neighbour solutions which are obtained by slightly changing the current solution (single move). A generic explained pseudo-code to HC to optimize a function \( f(x) \) is listed:

```plaintext
// Generate initial random solution and assign it to starting solution.
startSolution=generateRandomSolution();

// Loop start.
loop {
```
// Finding set of neighbours to current solution.

    Neighbours=findNeighbours(currentSolution);

// Reset optimization value and next best solution.

    nextVal=0;

    nextSolution=NULL;

// Loop in neighbours set to find the best neighbour solution.

    for all n in Neighbours
        if (f(n)>nextVal)
            nextSolution=n;
            nextVal=f(n);

    // Compare best neighbour to current solution.
    if (nextVal<=f(currentSolution))
        // Return current solution since no better neighbour solution exist.
        return currentSolution;

    // Otherwise take the best neighbour as the current solution.
    currentSolution=nextSolution;

}// Loop.

Contract-Net approach (Market-based)

While in CNP all workers enter auctions to bid for unscheduled tasks and the highest bidder schedules the task. Auctions are lead by an auctioneer calling for proposals (CFP’s) in sealed bids from workers, each worker bids with its marginal utility value (if not negative) of best scheduling the task in its schedule, the highest bidder wins the round, schedules the task, and wait for next rounds. Workers reject CFPs if the new
task won’t raise their utility, or their schedules are already full. The worker’s utility function used is described as follows:

\[ u_x = \alpha B_x - \beta \text{Dis} \text{tan } ce(T_{x-1} : T_x) \]

Respecting: \[ D_x = D_{Ex} + \text{Travel}(T_{x-1} : T_x) \leq \text{ScheduleFreeTime} \]

Where:

\( u_x \): Utility gained from scheduling task X.

\( B_x \): Bonus points coupled with task X.

\( \text{Dis} \text{tan } ce(T_{x-1} : T_x) \): distance from location of task \( T_{x-1} \) to \( T_x \) location.

\( \alpha \) and \( \beta \): Conversion operands, in our model \( \alpha = \beta = 1 \).

3.2 Schedule execution

After schedules were generated, workers enter execution phase during which multiple kinds of exceptions are generated (travel, setup, and execution exceptions) with different degrees (predefined or unknown delays, cancellations) according to a suitable probability model and introduced as discrete events in the simulation environment. Timing is also important to synchronize the workers; therefore this phase must be a conjunction of Discrete Event System (DSE) and time slicing simulation.

Exception handling strategies are executed in this phase to deal with uncertainty.

4. Initial Results:

Simulator’s planning phase has been implemented and tested. We ran multiple rounds on different sizes of our MTAP, and on the average, the centralized approach showed better performance of about 10% over the market-based one.
5. Future Work:

The execution phase of the simulator is being designed. Environment’s uncertainty is being investigated how to be modelled statistically. Once the simulator completed, multiple rounds will be run for both approaches with different scenarios and input values, then multiple analysis will be applied on the gathered data.

References:


