

A Hybrid Heuristic Approach to Provider Selection and Task Allocation Problem in Telecommunications with Varying QoS Levels

Nihat Kasap¹, Berna Tektaş Sivrikaya², Hasan Hüseyin Turan³, Dursun Delen^{4,*}

¹ School of Management, Sabanci University
Tuzla, Istanbul, 34956, Turkey
E-mail: nihatk@sabanciuniv.edu

² Department of Business Administration, Izmir Katip Celebi University
Cigli, Izmir, 35620, Turkey
E-mail: berna.tektassivrikaya@ikc.edu.tr

³ Department of Mechanical and Industrial Engineering, Qatar University
Doha, 2713, Qatar
E-mail: hasan.turan@qu.edu.qa

⁴ Spears School of Business, Oklahoma State University
Stillwater, Oklahoma, 74078, USA
E-mail: dursun.delen@okstate.edu

Received 14 August 2016

Accepted 26 March 2017

Abstract

In this research, we study a cost minimization problem for a firm that acquires capacity from providers to accomplish daily operations on telecommunication networks. We model the related optimization problem considering quality of service and capacity requirements and offer a solution approach based on genetic algorithm (GA). Our model reckons the tradeoff between the network capacity acquisition cost and opportunity cost arise when data transmission quality for real-time tasks manifested at undesired levels. To better represent the related features and complexities, we model both capacity and loss probability requirements explicitly, and then, formulate delay and jitter requirements as level matching constraints. Using an experimental framework, we analyze how optimal behavior of the firm is affected by different price schemes, transmission quality and task distributions. We also compare three GA based heuristic solution approaches and comment on the suitability of them on resource selection and task allocation problems.

Keywords: Hybrid GA, telecommunications, provider selection, resource allocation, quality of service (QoS)

1. Introduction

Telecommunications industry has become more complex during the last couple of decades because of the increasing number and diversity of actors, their continuously changing business strategies and business models. Firms utilize telecommunication networks to

maintain their daily activities including traditional business transactions and real-time applications such as audio/video streaming and voice over TCP/IP. Each of these applications may have different network capacity, due date and quality of service (QoS) requirements.

Firms acquire network capacity (a certain level of bandwidth) required to carry out their daily operations.

* Corresponding author: Phone: +1 918-594-8283; Fax: +1 918-594-8281

This capacity can be acquired from network operators such as AT&T and SPRINT, or from brokers that offer bandwidth polled from multiple network/backbone providers. While the backbone providers own the network, brokers do not have their own network infrastructure. In order to provide services broker firms must lease capacity from the network operators, and sell it to consumers in variety of configurations to make the margin. For example, Bandwidth Brokers Group, (<http://bandwidthbrokersgroup.com/about-bandwidth-brokers>) sells bandwidth either as a bundle or individually. Giglinx Global (<http://www.giglinx.com/tier-1-bandwidth-brokers.html>) is another bandwidth broker offering highly flexible packaging configurations by leasing network capacity from multiple globally distributed network providers, and by doing so successfully, able to offer competitive bandwidth pricing and premium support.

Companies (i.e., bandwidth consumers) obtain certain level of network capacity from bandwidth providers (which can be broker-type firms or genuine network operators with their own infrastructure) and then assign the acquired bandwidth to their planned activities.¹ Characterization of a network resource usually includes bandwidth capacity, time duration and QoS. In this study, as is the case in previous studies in this domain, we make the assumption that the contract and/or the service level agreement (SLA) signed between the consumer firm and the network provider firm specifies the level of bandwidth, QoS, and duration, and reports these metrics on a per period basis.

There usually are two different types of cost categories related to the use of telecommunication networks. The first one is the acquisition cost that is associated with the procurement of the network resources. The second one is the opportunity cost that incurred when transmission rates for real-time tasks drop below the desired levels. Since firms have a heterogeneous set of tasks (daily activities) with different bandwidth and QoS requirements, they either use more than one providers and/or sign different contracts specifying different bandwidth and QoS demands. In practice, bandwidth resources at lower QoS levels are often used for traditional low-volume/less-urgent data applications, while bandwidth resources with higher QoS levels (i.e., more expensive resources) are used for time sensitive/critical applications. Hence, firms are often in a position where they have to deal

with a cost minimization problem that exposes a tradeoff between resource acquisition cost and the opportunity cost.

We model the cost minimization problem from the bandwidth-consuming firm's point of view in a telecommunication network environment where capacity can be acquired from one or more providers, each with different QoS levels and with competitive price schemas. In some form or fashion (a combination that is the most advantages), the firm has to first obtain the needed network capacity (i.e., data bandwidth) from one or more providers, which may be bandwidth brokers or network operators (or a combination of both, or multiple contracts with one or more of the providers) to complete the planned tasks using the obtained capacity.

In telecommunication networks, the resource quality is common measured by delay, jitter and packet loss. Delay is the characterization of how long does it take for data to go through the network from a source to a destination. Jitter represents the variance in delay.² While real-time applications are very sensitive to delay and jitter, traditional data applications are not, (although, they are sensitive to packet/data loss). In video application, jitter causes intermittent freezing, and in audio applications it leads to awkward pauses in the middle of a conversation. Sudden changes in jitter may also cause packet loss.³ Generally speaking, in such situations where there is a packet loss, the receiver has two options: (1) completely ignore the lost data, or (2) send a request for retransmission to the sender. For real-time applications, which is also called time-fixed application such as videoconferencing, ignoring the loss of a packet may be more appropriate as these application are less insensitive to small number of losses. However, for data applications, which are also called size-fixed applications such as file transfer, retransmission is more appropriate, since the lost packet may invalidate the completeness and accuracy of the transmitted content.

QoS of the resources may affect the bandwidth consuming firms in two crucial ways. First, size-fixed tasks might use too much capacity, since they need to be retransmitted at the event of packet loss. Second, opportunity cost can be realized by decreasing transmission rate when there is unacceptable levels of quality for a time-fixed task such as a videoconference. Opportunity cost reveals the importance of a task since

the importance of a task can be correlated with higher penalty for not achieving the desired level of audio and video quality. While assigning tasks to resources, decision maker aims to minimize the total cost by consider the fine balance/trade-off among these cost items.

In this study, we model the consuming firms' situation as an optimization problem where the objective is to minimize the cost while satisfying the QoS and bandwidth capacity requirements (constraints) based on known tasks to be completed during a single planning period. Our model takes into account the trade-off between the capacity acquisition and the opportunity cost. The rest of this paper is organized as follows: Section 2 briefly describes the most relevant studies published in extant literature, by doing so, justifying the value and novelty of this study. Section 3 provides specification and nomenclature of our proposed model along with the modeling assumptions. Section 4 is dedicated to the explanation of our GA-based heuristic solution methodology. Section 5 offers the details about the computational analyses and presents the results. Section 6 concludes the paper by providing a summary of our study and offering some future research directions.

2. Literature Review

To lay the foundation for our research effort, we classify and analyze the existing literature into two groups: application domain specific studies and heuristic method related studies. QoS and pricing related managerial and infrastructure level researches are listed and discussed in the first group while the studies related to Genetic Algorithm (GA) heuristic search technique are summarized in the second group. The review section is then conclude with differentiating and signifying out study in the light of the previous literature.

2.1. QoS and pricing related studies

One branch of the first group considers network infrastructure design to guarantee QoS level through bandwidth allocation and operating policies. There are numerous studies on these issues in literature that represents analysis of packet scheduling techniques and algorithms to mitigate loss of data/information, buffer/temporary storage algorithms, and variety of service delivery models.^{4,5,6,7,8,9}

The other branch in the first group of the extant literature relate to the supply side concerns such as pricing schemas and QoS offerings and combination of the two. Gupta et al.¹⁰ claim that in order to improve the implementation and applicability, pricing should not be set at the packet level, rather multiple priorities with different QoS levels are needed to handle pricing applications easily and inexpensively. Many pricing strategies have been studied in the literature, of which the two that stands out at a high level are flat-rate pricing for each and every service,¹¹ and priority pricing where a variety of service classes are created and offered, each having varying levels of quality performance and price tags.^{12,13} In highly cited studies, Altmann and Chu,¹⁴ and Mackie-Mason and Varian¹⁵ model two layers of pricing: a flat-rate pricing is offered for basic services, and a usage-based pricing offered for higher quality services and tasks. In a dual mode pricing schema, they have shown that congestion type capacity problems can be mitigated by the proper application of usage-based pricing schema. Courcoubetis et al.^{16,17} offer variety of pricing schemas based on the effectiveness of the bandwidth delivery as the basis for usage and related costs/charge. In order to operationalize their pricing model, they proposed QoS constraints such as delay and packet loss as the key factors for pricing models. They postulate that because of the complexity and to ease the burden on applicability, the bandwidth providers should offer QoS differentiated pricing services at most two QoS levels.

There are several studies in the literature related to provider selection and task allocation problem in Telecommunications with customer side concerns. Pan et al.¹⁸ studies a fuzzy multi-objective optimization model for solving the provider selection problem, considering non-linear objective membership function, price breaks, different penalty and QoS levels in different tasks. Kasap et al.¹⁹ shows that their proposed heuristic approach with QoS degradations during the allocation of tasks can lead to more promising results especially while applying certain cost penalty policies to the reduction of QoS requirements. Turan et al.²⁰ studies provider selection and task allocation problem considering random QoS variables to capture stochastic nature of Telecommunications environment. Although, closely-related models have been studied and some efficient heuristics have been proposed in articles mentioned above, using GA based hybrid heuristics and

finding better outcomes is the added value for our study compared with the heuristics in studies mentioned above.^{1, 18, 19, 20, 21}

2.2. GA related studies

There are many analytic studies in the literature using GA as a technique to solve highly complex decision problems. It is arguably the most popular and therefore the most commonly used heuristic algorithm available in analytics. GA is a biologically inspired random search algorithm that can potentially find the optimal (or near-optimal) solution in complex multi-dimensional

search algorithm that has its own set of parameters (e.g., population size, crossover rate, mutation rate, etc.) prone to their own “optimal” search processes. Some of the best-practices-driven GA parameters and their plausible values are given in Table 1.

Another issue in proper design and implementation of GA is how to identify a suitable set of values for the genetic operators – selection probability, crossover rate and mutation rate. Usually the methods and related values used for selection of parents from the current population and selection of subsequent members of the next generations are different.

The aim of the parent selection procedure is to find

Table 1. Suggested GA parameters

Parameters	Negnevitsky	D. Jong	Michalewicz	Schaffer	Grefenstette
Population Size	50	50-100	50 -100	20-30	30
Crossover Rate	0.7	0.6	0.5 -1.0	0.75-0.95	0.95
Mutation Rate	0.001-0.01	0.001	0.001- 0.01	0.005-0.01	0.01

search problems.²² The algorithmic extend of GA is modeled after the natural evaluation of species where the operators that it employs are all inspired by the natural evaluation process. The GA operators, also known as genetic operators—selection, crossover and mutation— are used to manipulate entities (i.e., individual species in natural world and potential solution alternatives in analytics world) in a collection of solutions (i.e., population) over many iterations (i.e., generations) to reach the ultimate optimality by gradually improving the objective function value (i.e., the overall fitness). GA provides a symbolic parallel to the original problem, That is, it does not know (or should know) the underlying content/knowledge about the search-based problem. Instead, it tries to optimize the outcome by dealing with a pseudo representation of the actual parameter space using the analogy of chromosomes and genes.²³

Execution of GA starts with a randomly or semi-randomly generated population of potential solutions.²² The population size in GA is denoted with N, and the value of N should be decided prior to start of processing. Each solution is represented by chromosomes that are made of units called genes, each of which encodes a particular variable (e.g., feature of the organism). The position of a gene within a chromosome also reflects on particular sequence of the variable list (e.g., feature of the organism).²⁴ GA is a

the ones that are more likely to reproduce more instances/offspring whose fitness value are higher than those of the ones produced previously. The selection of the individuals can be indicated as a two-step process: (i) determination of number of trials an individual can expect to receive; (ii) conversion of the expected number of trials into a discrete number of offspring.²⁵ Several number of selection methods can be applied to GA including random selection, roulette wheel selection, ranking selection, stochastic universal sampling and elitism, etc. Proportional selection, which is called “roulette-wheel” selection, is the most used standard selection operator.^{25,26} Fitness values of individuals proportionally corresponds to the size of the space allocated to them on an imaginary wheel;²² larger the space, higher the chances of being selected. That is, even the most dominant individuals (as per their fitness values) have a decent chance of being left out, Fitness in natural systems is often interpreted as the ability to survive and multiply, as such, representing the natural/random selection process observed in natural world. This kind of random selection prevents the some of the technical problems such as premature (i.e., local minimum/maximum; or sub-optimal solution) convergence.²⁷

The crossover operation is widely considered the feature that makes GA significantly different than many of the other heuristic search algorithms, such as tabu

search, simulated annealing and dynamic programming.²² The crossover operator is used to create two new offspring solutions (children) using two existing solution alternatives (parents) that are selected from the current population by the selection operation. Some of the most commonly practice typed of crossover operations include one-point,^{28,29} two-point, multiple-point crossover,²⁶ cycle and uniform crossover.^{30,31,32}

In the mutation procedure, all potential solutions in the population are processed one-by-one and bit-by-bit, and the bit values are either reversed or kept as is using a random chance calculation defined by the predetermined mutation rate. The mutation operator is designed to help GA prevent the process form premature/sub-optimal convergence by forcing the algorithm to venture into new solution spaces. As opposed to routinely applying mutation operation, it is better to design and implement “smart” mutation strategies. One of these strategies promotes manipulating the mutation rate according to the evolution trend: increasing the mutation rate when a stalemate situation is reached, and inversely, decreasing the mutation rate when the solution fitness values are showing chaotic behavior, presenting values that are all over the probable range. Another strategy suggests adjusting the mutation rate according to the fitness value of a chromosome. Similarly, yet another strategy exemplifies the previous strategy but also prescribes different mutation probabilities for different parts of a chromosome.²² Using similar smart strategies, several researchers successfully applies GA based solutions to different application domains, such as Jin et al.³³ applies GA for the first time to the multi-stage hybrid flow job shop scheduling problem, and Gul et al.³⁴ applies it to scheduling of surgical services.

In conclusion, even though there exist several research articles related to modeling and implementation of QoS measures as given above, some of them even including GA applications, in the telecommunications as well as other areas, to the best of our knowledge there is no other study that uniquely combines and leverages both of these concepts in telecommunications management research. Moreover, proposed research differs from current literature in the following manners: (1) modeling of QoS parameters from firms' point of view rather than service providers or network owners (suppliers), and (2) not only modeling and solving bandwidth sourcing problem of

firm but also providing decision support mechanism for task allocation issues by integrating GA techniques.

3. Development of the Model for Provider Selection and Task Allocation

In this study, we consider telecommunication environment at which multiple providers exists and the firm can acquire network capacity (bandwidth) with different QoS levels at different prices. Similar to the model developed by Kasap et al.¹ and Zhou et al.²⁹, we model the capacity and the packet loss probability requirements explicitly, but formulate delay and jitter as level matching constraints. We also consider the all-you-can-send pricing scheme¹⁶ in which the firm gives a fixed price for a fixed bandwidth available for a fixed duration.

Generally speaking, there are two different types of tasks that use data networks. A real-time task which is *time-fixed* since its size can be compressible and changed without affecting its completion but the transmission duration cannot be changed. Examples of such includes audio/video transmission tasks such as teleconferencing and video/audio streaming. A task is *size-fixed* if all bits have to be transmitted at the same time but the duration can change. Most traditional data applications such as file transfer, database transactions are considered to be size-fixed. This two-task type classification provides a reasonable representation of the common practice in real-life applications.¹

3.1. Problem notation

In order to maintain consistency, understandability and resemblance to common practice as they are covered in the related literature, from heretofore, we will use the following nomenclature (i.e., representation of parameters and decision variables) while presenting formulations and solution procedures in this paper.

Parameters:

- I, J : The ordered index set of resources, and tasks respectively.
- A_T, A_S : The index set of tasks with fixed transmission time and fixed size respectively where $A_T \cap A_S = \emptyset$.
- q_j : The minimum quality level tolerated by task j . $q_j = q(\delta_j, \sigma_j)$ δ_j, σ_j are delay and jitter for task j , respectively.

- Q_i : The quality level guaranteed by resource i .
 $Q_i = Q(\delta_i, \sigma_i)$ where δ_i, σ_i are delay and jitter levels of resource i , respectively.
- α_i : Transmission efficiency, calculated as one minus the packet loss rate of resource i .
- β_i, L_i : The bandwidth and duration of resource i . $L_i = \min(\text{length of contract, planning horizon})$
- c_i : The total cost of resource i for specific β_i, L_i .
- c_j^o : The opportunity cost of missing the target transmission rate for task j .
- \bar{r}_j^U, \bar{r}_j^L : Target and the minimum transmission rate of task j at the receiving node, respectively.
- Δt_j : Estimated scheduled transmission time for time-fixed task $j \in A_T$.
- \bar{x}_j : The (fixed) length of task $j \in A_S$ in number of bits.

Decision variables:

- v_i : 1 if resource i is selected, zero otherwise.
- r_j : The transmission rate of task j .
- y_{ij} : 1 if task j is assigned to resource i , zero otherwise.
- y_{ijt} : 1 if task j is active (transmitting) at time t on resource i , 0 otherwise.
- t_j : Start time of task j .

3.2. Problem formulation (Problem P1)

$$\text{Min } z = \sum_i c_i v_i + \sum_i \sum_{j \in A_T} (\bar{r}_j^U - r_j \alpha_i) c_j^o y_{ij} \quad (1)$$

Subject to

$$\sum_{j \in A_S} \bar{x}_j y_{ij} + \sum_{j \in A_T} \alpha_i \Delta t_j r_j y_{ij} \leq \alpha_i \beta_i L_i \quad \forall i \in I \quad (2)$$

$$q_j \geq Q_i y_{ij} \quad \forall j \in J, \forall i \in I \quad (3)$$

$$(t_j + \Delta t_j + \delta_i) y_{ij} \leq L_i \quad \forall j \in A_T, \forall i \in I \quad (4)$$

$$\sum_{j \in J} r_j y_{ijt} \leq \beta_i \quad \forall i \in I, t = 1..L_i \quad (5)$$

$$\sum_{i \in I} y_{ij} = 1, \quad \forall j \in J \quad (6)$$

$$\sum_{i \in I} \sum_{t \leq L_i} y_{ijt} = \Delta t_j, \quad \forall j \in A_T \quad (7)$$

$$\sum_{i \in I} \sum_{t \leq L_i} r_j y_{ijt} = \bar{x}_j, \quad \forall j \in A_S \quad (8)$$

$$y_{ij} \leq v_i \quad \forall i \in I, j \in J \quad (9)$$

$$y_{ijt} \leq y_{ij}, \forall i \in I, j \in J, t \leq L_i \quad (10)$$

$$r_j \alpha_i \geq y_{ij} \bar{r}_j^L \quad \forall j \in A_T, \forall i \in I \quad (11)$$

$$r_j \alpha_i y_{ij} \leq \bar{r}_j^U \quad \forall j \in A_T, \forall i \in I \quad (12)$$

$$r_j, t_j \geq 0$$

$$v_i, y_{ij}, y_{ijt} \in \{0, 1\} \quad (13)$$

Problem P1 considers the tradeoff between the capacity acquisition cost and opportunity cost due to not meeting target transmission rates in real-time tasks. Constraint set (2) guarantees that we can use only up to available capacity. The QoS requirements of the resources satisfying the minimum QoS requirements of the tasks that are assigned to them are ensured by Constraint set (3). Constraint (4) ensures that all time-fixed tasks assigned to a resource are completed when the resource is available (time dimension). Moreover, a similar constraint is not necessary for size-fixed tasks since Constraint set (2) guarantees that a size-fixed task is only assigned if there is enough capacity. Constraint set (5) prevents using more bandwidth than available at any time (bandwidth dimension). Constraint (6) along with (13) ensures that a task is assigned to only one resource and all tasks are assigned. Constraint set (7) and (8) guarantees that the tasks are actually allocated the required amount of time slices. Constraint set (9) guarantees that a network resource is selected only if at least one task is assigned to it. Constraint set (10) ensures that a task is assigned to a network resource only if it occupies a time slice on it. Constraint set (11) states that transmission rate for a time-fixed task j should be high enough to satisfy the minimum transmission (reception rate) at the sink node. Constraint set (12) enforces the target transmission limitation for all tasks.

Figure 1 conceptualizes P1 as fitting flexible items in a two-dimensional bin. Bin dimensions are raw bandwidth, β_i , and the duration of resources L_i . Tasks can be conceptualized as two-dimensional shapes. The x-axis and y-axis represent usage time (transmission time) and transmission rate respectively and the area of the shapes represents the size of the tasks.

Unlike a bin-packing problem (BPP) P1 allows, as Figure 1 illustrates, the duration of the tasks in A_T to be fixed. Since the transmission rate can vary, the size of the task can change. Moreover, the areas of the tasks in A_S are fixed but the shapes are variable since the transmission rate and the transmission time can both

vary as long as the size remains fixed. The shape of items in the set A_S could be a single rectangle or a combination of several rectangles (item 5 in Figure 1).

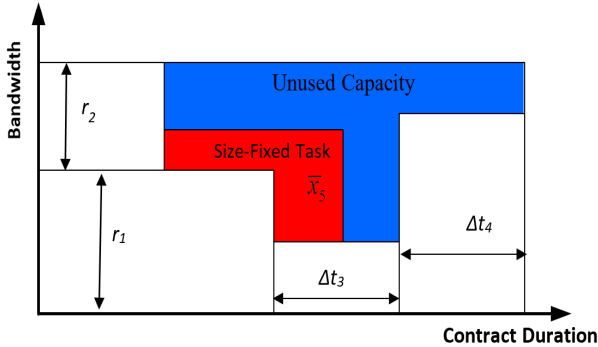


Fig. 1. A schematic representation of resources

At any time, the summation of transmission rates of active tasks has to be less than or equal to the bandwidth of resources.

The problem is NP-hard in the strong sense.³⁵ Although obtaining an exact solution to the problem is difficult (if not impossible), a meta-heuristic that yields satisfactory sub-optimal solutions can be formulated. In the next section we present our GA-based heuristic approach to this problem. The flow chart for the proposed GA based hybrid heuristics is depicted in Figure 2.

4. Solution Procedure Based on Hybrid GA Approach

As a solution procedure we offer three meta-heuristics combination of GA approach and the heuristic developed by Kasap et al.¹ and Zhou et al.²⁹ In the first two meta-heuristics (GA1 and GA2), the tasks are assigned to resources by using GA approach. After assignment, start time and transmission rate of each tasks (in other words task allocations) are handled by using the idea of the heuristic (called Heuristic A-that is used in steps 3 and 5 in Figure 2) developed by Kasap et al.¹ In addition, after completing all iterations in GA2, we call Heuristic A so that we can use resource reduction idea of it.

Hence, we expect to have better results compared to GA1. In the third meta-heuristic (GA3), we use GA approach only to identify which resources will be selected. After that, task assignments to these resources

and start time and transmission rate of each task in other word task allocations are handled by using the idea of Heuristic A. Since a GA solution requires a proper

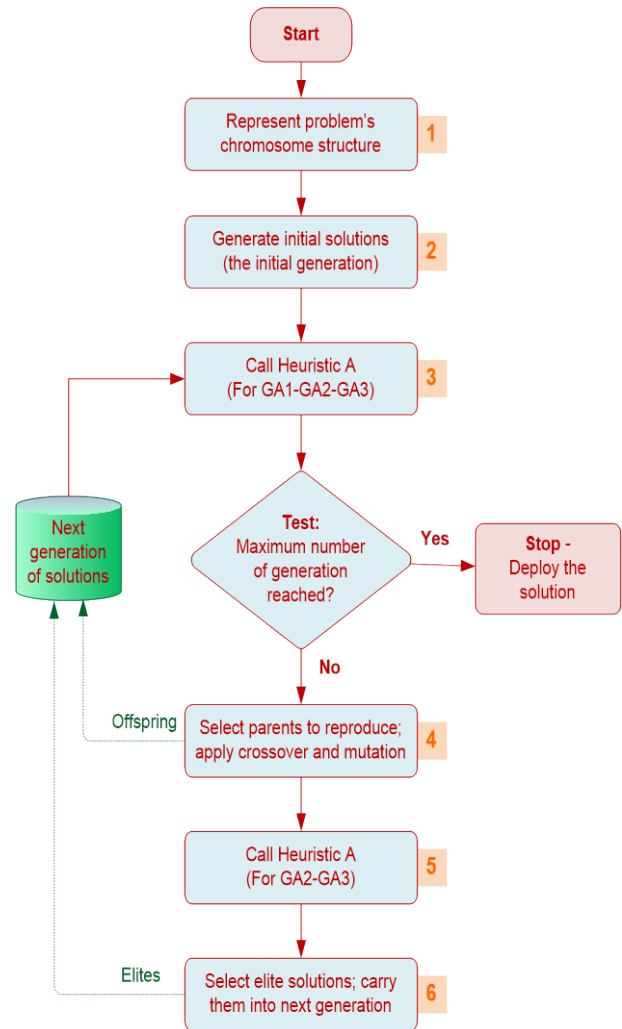


Fig. 2. Flow chart for the proposed GA based Hybrid Heuristics

representation of chromosomes/genes, along with the identification of GA operator values, we dedicate the next sub-section to explain the nature and reasoning of our decisions.

4.1. Designing chromosome and initial population in GA

According to model, a task can be allocated to only one resource, however more than one task can be allocated in a leased network capacity (resource). Moreover, a feasible sub-set of resources from the available resource

set can be acquired, but all of the scheduled tasks have to be completed, which means all the tasks should be allocated to a resource. Hence, we designed our chromosomes (for GA1 and GA2), by adopting an integer permutation similar to some studies in the literature.²⁹ The genetic encoding of an m -task and r -resource allocation problem can be stated in a scheme based on an m -dimensional matrix where the position of each gene in the chromosome indicates the number of a task, j , as in Figure 3. Therefore, the length of a chromosome is equal to the number of the tasks, m , to be assigned to the resources, and we coded chromosomes with real numbers indicating a specific resource (showing which resource task allocated in) from the resource set. This genetic representation

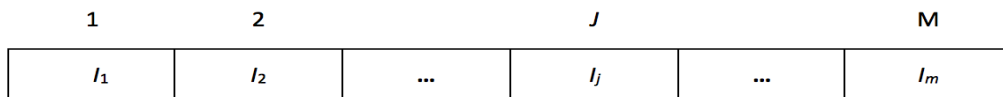


Fig. 3. A chromosome representation for provider selection and task allocation problem

scheme ensures that equality constraint in (6) is automatically satisfied, since only one resource is assigned to each task. This stage corresponds to step 1 in Figure 2.

In this study, we use a population size of 100, $K = 100$,^{28,31,32,36} running for 1000 generations for GA1 and GA2). We also use a population size of 40, $K = 40$, running for 100 generations for GA3 since running time of GA3 is too much compared to GA1 and GA2 (see step 2 in Figure 2). Hence chromosomes contain binary values in GA3 since each gene in the chromosome indicates whether the resource is selected to use or not.

We create the initial population of possible solutions by using both randomly produced solutions and initial solution of Heuristic A. While producing random solutions we consider unit cost, total cost and capacity of resources and assign different weights to them. In other words, we do not give equal chance to resources to be selected. First we order resources according to their unit costs in ascending order. We assign higher chance to cheaper resources. Second, we order resources according to their total costs in ascending order. We assign higher chance to cheaper resources. Third we order resources according to their capacity in descending order. We assign higher chance to larger resources. We then randomly produce three subset of initial population considering these chances. Moreover,

we consider total task size in other words minimum required total capacity and give more chance to sorted resources whenever their overall capacity is equal or less than total task size.

Heuristic A produces initial solutions for P1 by setting the transmission rate of time-fixed tasks to their upper bounds. In the case where results of Heuristic A is used as an initial population, the GA starts the optimization with a set of approximately known solutions and therefore converges to an optimal solution in less time than using only randomly produced solutions. However, the size of time-fixed tasks can be changed without disrupting its completion and, there can be better solutions which have an opportunity cost because of size compressible feature of time-fixed tasks.

Therefore, if only the initial solutions of Heuristic A are used, the GA has the risk of converging to local optima. Oppositely, if we use only random solutions, the CPU time of the GA can be too long. Because of aforementioned reasons, in our tests, we create 25% and 40% of the initial population by using Heuristic A, and the remaining population individuals by using a random number generator.

When a traditional GA is applied to the problem under investigation, it produces numerous infeasible chromosomes in each generation, resulting poor performances. At each iteration, for each possible solution (in other words for each chromosome), start time and transmission rate of each tasks are calculated (in other words tasks are allocated to given resources) by using the allocation idea of the Heuristic A, which corresponds to step 3 in Figure 2. The fitness of each chromosome is computed as the objective function value of the model. Whenever, the heuristics cannot allocate tasks to given resources, we can say that the solution is infeasible and we can assign very high objective function value. Since the GA algorithm selects the best solutions at every iteration as the next population, it never selects infeasible solutions.

4.2. Selection

In GA approach, first the chromosomes that are used in the crossover operation have to be selected. In the selection operation, elitism method is applied since it first determines chromosomes with the best fitness value, and then transfers their raw fitness values into the next generation in order to create more chromosomes with higher fitness values into the entire population of the next generation.

We employ linear ranking selection to individuals. The population is sorted from the best to the worst chromosomes and the selection probability of each chromosome depends on the given rank.²⁴ The ranking function has the following form:

$p_j = q - (j - 1)r$ where p_j is the selection probability for the j th chromosome in the ranking of the considered population, q is the probability of the best chromosome and q_0 the probability for the worst chromosome we can determine the parameter r as follows:²⁴

$$r = \frac{q - q_0}{K - 1} \quad \text{where } K \text{ is the population size.}$$

Intermediate chromosomes' ranks are decreased from q to q_0 proportionally to their rank.

$$\text{We set } q_0 = \frac{1}{K(K+1)/2}, \quad \text{and } q = \frac{K}{K(K+1)/2},$$

therefore, r becomes $\frac{1}{K(K+1)/2}$.

4.3. Crossover

In this study, 0.6 and 0.8 crossover rates, that are compatible with suggested rates given in Table 1, are used during the computational analysis of GA. In order to find parents in the crossover operation, we use linear

ranking idea as in Section 4.2. We use a one-point crossover operator, by which two parent chromosomes are each cut into two segments and two new chromosomes are obtained by concatenating these segments (Figure 4).

Since the process is totally random, a newly created solution (i.e., chromosome, which may be a member of the initial population or created as part of mutation/crossover operators) may not readily match to a feasible resource allocation schema. In this study, in order to maintain richness of the solutions space, we allowed for infeasible chromosomes, however, after identification of an infeasible chromosome, we use the allocation step of the Heuristic A and accordingly assign fitness value. That is, and as explained in Section 4.1, we assign high fitness value to infeasible chromosomes so that they will not be selected for the next generations.

4.4. Mutation

Crossover and mutation correspond to step 4 in Figure 2 and numeric examples for this step are given in Figures 4 and 5. In GA, the mutation operator randomly selects a gene (representing a specific task) and then changes value of that gene which implies that the task is allocated to a different resource in the resource set, I (Figure 5). This new resource of the task j can be either a resource used by other tasks or completely a new resource that is not used before by any task. In order to eliminate assigning completely new (not used before for assignment) resource, we modify traditional mutation function. If the randomly generated value for the gene in mutation represents empty resource (not used before), we enforce generation another value for gene until non

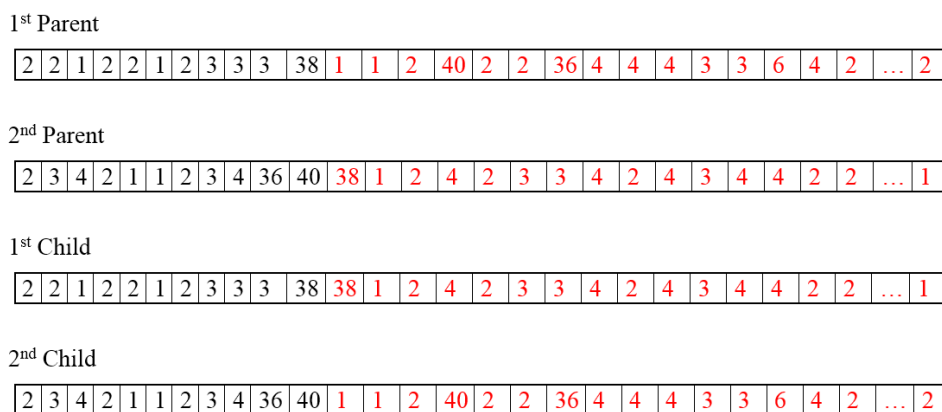


Fig. 4. Crossover operation

empty resource is selected.

We also modify gene selection step in mutation. We assign high probability for the genes, whose representing resource have higher available unused capacity (in other word resource utilized less), to be selected for the mutation. Finally, for the mutated value

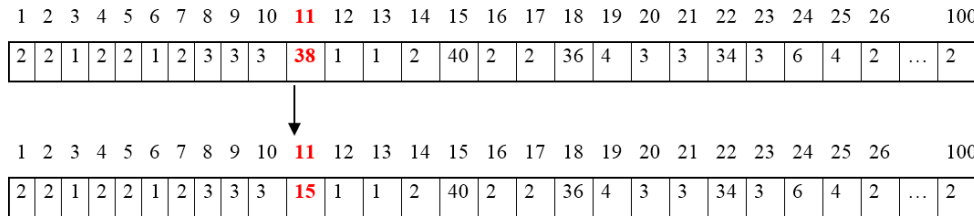


Fig. 5. Real number uniform mutation

of the gene, we assign low probability for the resources whose available unused capacity is high. By modifying mutation operator, we help closing (not selecting) resources that are utilized less. In this study, we use mutation rate as 0.01.

4.5. Replacement and elitism

In GA the size of population remains the same throughout iterations. Therefore, after crossover and mutation, some of the chromosomes have to be eliminated. We sort chromosomes based on their fitness value. Then we select best K chromosomes for the next generation. Elitism method determines chromosome with the best fitness value, and then transfers its raw fitness value into the next generation in order to create more chromosomes with higher fitness values into the entire population of the next generation. Elitism correspond to step 6 in Figure 2.

5. Computational Results for GA Approach

The data set that we used in this study contains 720 problems with 36 different problem settings. As shown in the Table 2, for the experimental design we use four factors/parameters to generate the 36 different problem settings/scenarios. All tests are performed on a 2.40 GHz Intel Core 2 quad CPU Q6600 processor with 4 GB RAM, running 32-bit Windows.

Three pricing scenarios (one random and two provider-sensitive) are used. In the first one, all resource usage costs are randomly generated. For the provider sensitive cases (Figure 6), the resource usage cost is

defined as a function of the bandwidth and the QoS level of the resource. In the parallel cost curve case, there is a dominant provider (i.e., the least expensive). In the other case, the price curves of different providers intersect each other so that at some bandwidth and QoS level a provider becomes more expensive. Intuitively,

the firm will use the lowest cost as long as the QoS constraints are satisfied. In cases where QoS cannot be satisfied the firm will switch to a competitor. We study the random cost structure solely to test the performance of the heuristic.

Table 2. Full-factorial experimental design

FACTOR	LEVEL
Pricing Type	3
Tightness	3
# Tasks Per Resource	2
# TF Tasks / # SF Tasks	2
Total # Problem Type	36

Another factor for the problem is tightness that is defined as the ratio of the total task size to the total resource capacity. Three levels (50%, 70%, and 90%) of tightness have been used in the computational study. It is expected that as tightness increases the number of resources used and the number of crashed tasks will increase. Two levels of average number of tasks per resource are 10 and 25. This is obtained by setting the ratio of average task size to average resource capacity to 0.1 and 0.04 respectively. We randomized sizes of tasks, resource capacities and QoS requirements while maintaining the above ratios. The last factor is the ratio of the number of time-fixed tasks to that of the size-fixed tasks and it varies from 1 to 2.

In order to test the solution quality of GA1, we run whole data set (720 problems) 8 times by setting 2 different crossover rate (0.6 and 0.8), Heuristic A initial solution percentage (25% and 40%) to generate population for GA1, and total task size coefficient (1.25

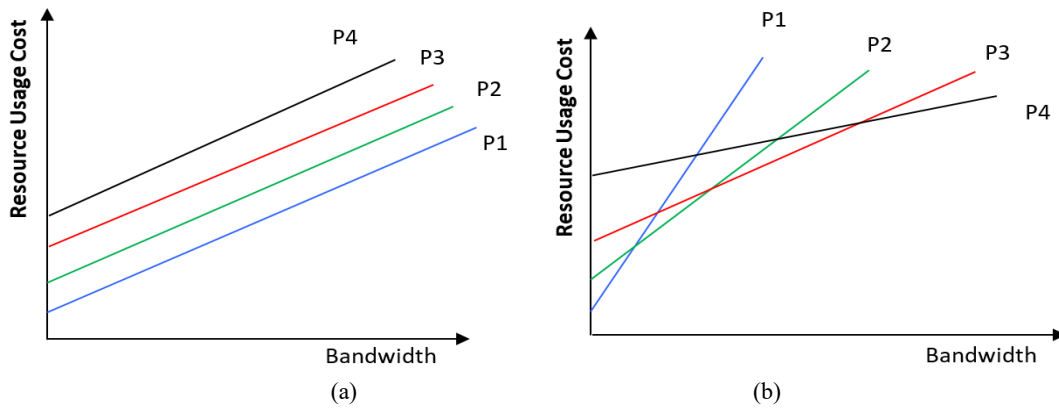


Fig. 6. Provider sensitive pricing, (a) parallel & (b) intersecting curves

and 1.5) showing minimum required total capacity. We also set population size as 100, mutation rate as 0.01, number of iteration as 1000 and number of crossover cutting point as 1.

Table 3 lists the average quality of solution for each problem setting. In the table, the column labeled as Z_{GA1}/LB shows the ratio of the GA approach outcome to the lower bound (LB). The overall average gap between solutions of GA1 and LB is 23.93% with minimum gap of 10.64%.

Table 3. Average Solution quality for GA1

Crossover Rate	Heuristic Solution Percentage	Total Task Size Coefficient	Average Z_{GA1}/LB
0.6	25	1.25	1.2393
0.6	25	1.50	1.2441
0.6	40	1.25	1.2451
0.6	40	1.50	1.2436
0.8	25	1.25	1.2452
0.8	25	1.50	1.2439
0.8	40	1.25	1.2433
0.8	40	1.50	1.2423

Although Heuristic A can reduce the number of resources used by crashing time-fixed tasks and swapping among resources, it may not produce optimal results. That is, the main idea behind the heuristic is to reduce the number of resources acquired (reducing the cost of acquiring capacity) by reducing (crashing) the transmission rate of time-fixed tasks just enough (creating opportunity cost) so that other unassigned

tasks can now use the slack capacity created by the reduction. Heuristic swaps the tasks among the resources so as to fit all of them. When sufficient slack capacity is created, one of the used resources is eliminated (Reduce the number of used resources) and the procedure is repeated until no slack capacity can be created. However, GA approach can find suitable resources used but could not eliminate one of them by crashing and swapping. In other words, it is suitable for resource selection but not task allocation since it requires too many swapping (it may require too many iteration).

As given in Gul et al.³⁴, using more computational effort with GA-based method under a restricted environment does not accomplish significant additional improvements. When we check number of resources used, GA1 uses 1-2 resources more than Heuristic A. Since cost of acquiring capacity is high enough, even one resource difference can create high Z_{GA1}/LB ratio.

In order to test whether GA is suitable for resource selection but not task allocation, we run GA3 and compare with GA1 and GA2. After completing all iterations in GA2, we call the Heuristic A so that we can use resource reduction idea of it (see step 5 in Figure 2). Hence, we expect to have better results compared to GA1. In GA3, we use GA approach only to identify which resources will be selected. After that task assignments to these resources and start time and transmission rate of each task in other word task allocations are handled by using the idea of the Heuristic A. Since GA3 uses task assignment and swapping ideas of this heuristic in every iteration, running time of GA3 is too much compared to GA1 and GA2.

In order to evaluate the effects of the experimental factors from Table 2, we use whole data set for the run (crossover rate = 0.6, Heuristic Solution Percentage = 25% and Total Task Size Coefficient = 1.25) having lowest Average Z_{GA1} /LB ratio in Table 3. The solution quality and run time comparison for all three GA heuristics is given in Table 4.

Table 4. Solution quality and running time comparison

Heuristic Type	Running Time (sec)			Average Z_{GA}/LB
	Min	Average	Max	
GA1	61.26	184.81	381.95	1.2393
GA2	59.03	314.93	1002.38	1.1128
GA3	27472	487523.14	2717560	1.0336

As we expected, Z_{GA2} /LB ratio is much better than Z_{GA1} /LB ratio since GA2 call Heuristic A at the end (after completing all iterations) and resource reduction idea is working well within a reasonable extra run time. Moreover, GA3 is having extremely good Z_{GA3} /LB ratio compared to other two heuristics and Heuristic A. The quality of GA approach is better than Heuristic A, since the overall average gap between solutions of GA3 and LB is 3.36% compared to 4.87%. Hence, GA approach is suitable for resource selection but not task allocation since it requires too many swapping. Since GA3 uses task allocation and swapping idea from Heuristic A, it requires to call Heuristic A for each population for each iteration hence it has too huge run time.

As given in Table 4, GA3 has run time on the average around 5 days with max value of around one month. That is why, we set iteration number as 100. If it is more than 100 run time could be too much, however, the solution quality could be better. Since we obtain reasonable better solutions compared to Heuristic A, we did not test with more iterations. The effect of pricing type is shown in Table 5.

The solution quality is low in case of provider sensitive pricing since only some of the resources are cheaper and GA approach need to find them as solution. However, in random pricing, there are many possibilities in the solution set so that solution quality is high. The most difficult problem type is the ones with intersecting cost curves. In parallel cost curve case there is a low cost provider at all levels of quality and bandwidth, however, in the third case where providers have intersecting cost curves, one provider is cheaper at low bandwidth, another is cheaper at high bandwidth and high quality. Finding best resource combination as solution becomes even more difficult. Since the difficulty level of problems increases with provider sensitive pricing especially intersecting cost curves and GA approach could not work with crashing and swapping to eliminate less used (utilized) resources, number of used resources increases. Run time for intersecting cost curve pricing is lower compared to other. GA approach works faster but could not find good solution compared to parallel cost curve case since intersecting cost curve problem type is difficult than others.

The effect of tightness is described in Table 6. When tightness increases from 50% to 90% the solution quality of GA1 decreases since it requires more swapping and GA approach is not good at task allocation and swapping. Since GA2 and GA3 uses task allocation, crashing and swapping idea from Heuristic A, the solution quality increases with fewer gaps to LB. When tightness increases, the difficulty level of problems increases. Moreover, GA2 and GA3 work excellent with more compact problems.

Table 7 represents the effect of number of tasks per resource. Since we kept the total required capacity constant across the scenarios the size of each task decreases as the number of tasks per resource increases. Hence, there is less of a need to crash tasks when the number of tasks per resource increases and GA1 solution quality increase. However, GA2 and GA3

Table 5. The effect of pricing type

Pricing Type	Run time (sec)					
	Average Z_{GA1}/LB	Average Z_{GA2}/LB	Average Z_{GA3}/LB	GA1	GA2	GA3
Random	1.1569	1.1593	1.0326	181.56	321.40	826800.49
Parallel cost curve	1.2226	1.0753	1.0312	186.46	315.01	339074.35
Intersecting cost curve	1.3384	1.1037	1.0370	186.41	308.37	296694.58

Table 6. Results based on tightness

Tightness (%)	Run time (sec)					
	Average ZGA1 /LB	Average ZGA2 /LB	Average ZGA3 /LB	GA1	GA2	GA3
50	1.1768	1.1602	1.0418	176.06	308.36	534612.18
70	1.2154	1.0964	1.0351	183.78	362.80	456627.51
90	1.3256	1.0818	1.0239	194.58	273.61	471329.72

Table 7. The effect of the number of tasks per resource

Number of Tasks per Resource	Run time (sec)					
	Average ZGA1 /LB	Average ZGA2 /LB	Average ZGA3 /LB	GA1	GA2	GA3
10	1.3033	1.0767	1.0312	73.99	136.69	82042.66
25	1.1753	1.1489	1.0360	295.63	493.16	893003.61

solution quality decrease since they use task allocation, crashing and swapping idea from Heuristic A more. Even solution quality decreases with increasing number of tasks per resource, the quality is still better than GA1. Moreover, since number of tasks per resource increases, it requires more calculations in GA1 and GA2. Hence run time increases with increasing number of tasks per resource. However, the increase in run time in GA3 is comparable less since this approach uses GA only resource selection not task allocation.

Table 8 documents the effect of the ratio of the number of time-fixed tasks to size-fixed tasks. When the ratio increases from 1 to 2, due to more time-fixed tasks, solution algorithm can perform more crashing and swapping since it is harder to schedule time-fixed tasks.

crashing and swapping for time-fixed tasks. Moreover, the ratio of time-fixed to size-fixed tasks may have an impact on the feasibility, since it is harder to schedule time-fixed tasks.

Table 9 represents the effects of pricing type on bin usage share. When there is random pricing, all providers have almost equal bin usage share as expected. In the parallel cost curve case, provider 1 with lowest cost has more usage share. Hence, in the intersecting cost curve case, provider 4 with lowest cost with high bandwidth has more usage share.

Table 10 represents the effect of tightness on opportunity cost, total cost and bin utilization. When tightness increases bin utilization increases. Since GA1 uses task allocation and swapping idea less, its solution

Table 8. The effect of TF/SF

Time-Fixed / Size-Fixed Task Ratio	Run time (sec)					
	Average ZGA1 /LB	Average ZGA2 /LB	Average ZGA3 /LB	GA1	GA2	GA3
1	1.2087	1.1079	1.0333	181.47	311.16	351337.99
2	1.2699	1.1177	1.0339	188.14	318.69	623708.28

Hence, GA approach solution quality decreases compared to Heuristic A. Since GA2 and GA3 uses task allocation idea from Heuristic A, their solution quality do not affected much compared to GA1 while considering the increase in the ratio of the number of time-fixed tasks to size-fixed tasks. Moreover, run time of GA3 increases a lot with increasing in the ratio of the number of time-fixed tasks to size-fixed tasks since it calls Heuristic A at every iteration and use more

quality is low and bin utilization values are really low compared to GA2 and GA3. Hence, it is better to compare GA2 and GA3. When tightness increases opportunity cost and total cost increases. It represents that the number of tasks crashed and take opportunity cost increases with increasing tightness. Hence, the number of resources selected increases with increasing tightness. Moreover, GA3 works better than GA2 since its total cost is low and bin utilization is high. It

represents that GA3 find good solution by using less resources and allocating task into them more compact.

Table 9. The effect of pricing type on bin usage share

Pricing Type	GA Type	% Bin Usage Share			
		P1	P2	P3	P4
Random		24.83	24.54	25.98	24.66
Parallel cost curve	GA1	62.55	30.78	6.30	0.36
Intersecting cost curve		0.29	1.19	17.61	80.91
Random		28.31	25.67	25.72	20.30
Parallel cost curve	GA2	70.48	25.01	4.28	0.22
Intersecting cost curve		0.36	0.87	12.86	85.91
Random		24.33	23.88	24.76	27.03
Parallel cost curve	GA3	65.15	27.55	5.09	2.21
Intersecting cost curve		0.00	0.00	8.77	91.23

Table 10. The effect of tightness on cost and bin utilization

Tightness (%)	GA Type	Opportunity Cost	Total Cost	% Bin Utilization
50	GA1	104.98	653.02	74.60
70		96.27	895.13	77.37
90		88.86	1238.34	78.08
50	GA2	111.18	524.95	95.74
70		113.13	706.33	98.01
90		117.83	918.12	98.66
50	GA3	50.01	479.67	99.32
70		75.02	680.39	99.62
90		107.36	868.23	99.46

As can be seen in Tables 11 and 12, number of tasks per resource and time-fixed / size-fixed task ratio has similar effect on opportunity cost and total cost comparing with tightness.

When number of tasks per resource increases opportunity cost and total cost increases. Hence, as time-fixed / size-fixed task ratio increases opportunity cost and total cost increases. Moreover, GA3 works better than GA2 since its total cost is low and bin utilization is high.

Table 11. The effect of number of tasks per resource on cost and bin utilization

Number of Tasks per Resource	GA Type	Opportunity Cost	Total Cost	% Bin Utilization
10	GA1	54.67	932.74	78.64
25		138.74	924.92	74.72
10	GA2	79.15	701.77	98.47
25		148.94	731.15	96.48
10	GA3	68.88	668.74	99.35
25		86.05	683.46	99.59

Table 12. The effect of time-fixed / size-fixed task ratio on cost and bin utilization

Time-Fixed / Size-Fixed Task Ratio	GA Type	Opportunity Cost	Total Cost	% Bin Utilization
1	GA1	80.67	878.57	79.71
2		112.74	979.09	73.65
1	GA2	95.88	704.64	97.40
2		132.21	728.29	97.55
1	GA3	60.39	655.79	99.46
2		94.54	696.41	99.47

6. Summary, Conclusions and Future Research Directions

In this paper, we presented a novel formulation of a cost minimization problem to solve a firm's network resource acquisition and task allocation problem subject to QoS requirements and opportunity costs. In our formulation, we modeled the packet loss probability requirements explicitly, and formulated delay and jitter as level matching constraints to simplify the model. Since the suggested model turned out to be NP-Hard complexity, obtaining an exact solution is difficult. In order to solve the proposed model, GA based meta-heuristics that yield good solutions were developed. The solution quality of heuristics was tested by provided lower bound method, which allows prorating and relaxes integer decision variable so that task splitting is allowed.

We suggested GA based heuristics that can take advantage of the trade-off between quality related costs and capacity costs. By comparing all three GA based heuristics, we can conclude that GA approach is suitable for resource selection but not task allocation since it requires too many swapping (it may require too many

iteration). As given in Gul et al.³⁴, using more computational effort with GA-based method under a restricted environment does not accomplish significant additional improvements within a reasonable time. Since task allocation requires too many swapping, we can integrate with good heuristics handling task allocation and swapping better such as Heuristic A. However, the disadvantage of calling Heuristic A is having too much run time since for each population and for all iterations we need to use this heuristic. However, GA3 obtains good results within fewer iteration since 100 is considered as less in GA algorithms. Finally, the quality of GA approach is better than Heuristic A, since the overall average gap between solutions of GA3 and LB is 3.36% compared to 4.87% in Heuristic A.

We demonstrated that the heterogeneous task set reveal some interesting challenges. We particularly focused on the all-you-can-send pricing where provider charges a fixed price for fixed bandwidth and duration. Computational study show how task distribution together with different prices and quality affect the optimal behavior of the firm. As tightness increases, the difficulty level of problems increases. Hence, GA2 and GA3 work excellent with more compact and difficult problems. It represents that GA3 find good solution by using less resources and allocating tasks into them more compact. However, GA approach works faster but could not find good solution in intersecting cost curve case compared to parallel cost curve case.

Although there are many studies in telecommunication sector about infrastructure and pricing issues in the literature, there are few studies about resource acquisition and task allocation strategies of firms combined with meta-heuristic solution. Hence, this study has important contribution to the literature to fills this gap.

From an application point of view, studying different pricing schemes could be suggested as future research direction since some pricing schemes will change the problem characteristics quite drastically and allow us to attain valuable managerial insights. Moreover, different meta-heuristics with GA such as Constraint Programming Based GA approach could be studied as future direction. Different chromosome and mutation structures could be tested.

Acknowledgements

This research is supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK), Career Development Program (Grant No. 106 K 263).

References

1. N. Kasap, H. Aytug and S. S. Erenguc, Provider selection and task allocation issues in networks with different QoS levels and all you can send pricing, *Decision Support Systems* 43 (2007) 375 – 389.
2. D. E. Comer, *Computer Networks and Internets with Internet Applications* (Third edn., New Jersey, Prentice Hall, 2001).
3. B. Teitelbaum and A. Sadagic, Internet2 QoS & Video: Sharing Responsibility to Overcome Congestion-Related Performance Problems, in *SURA/ViDe 4th Annual Digital Video Workshop* (University of Alabama, Birmingham, April 25th, 2002).
4. B. Carpenter and K. Nichols, Differentiated Services in the Internet, in *IBM Research Report* (RZ 3395 # 93445, 02/11/2002).
5. A. Habib, S. Fahmy, S. R. Avasarala, V. Prabhakar and B. Bhargava, On detecting service violations and bandwidth theft in QoS network domains, *Computer Communications* 26 (2003) 861-871.
6. B. Li, M. Hamdi, D. Jiang, X. R. Cao and Y. T. Hou, QoS-Enabled Voice Support in the Next-Generation Internet: Issues, Existing Approaches and Challenges, *IEEE Communications Magazine* (April 2000) 54-61.
7. R. Pletka, P. Droz and R. Haas, A New Buffer Management Scheme for IP Differentiated Services, in *IBM Research Report* (RZ 3216 # 93262, 03/13/2000).
8. S. Verma, R. K. Pankaj and A. L. Garcia, Call Admission and Resource Reservation for guaranteed Quality of Service (GQoS) Services in Internet, *Computer Communications* 21 (1998) 362-374.
9. M. C. Yuang and Y. R. Haung, Bandwidth Assignment Paradigms for Broadband Integrated Voice/Data Networks, *Computer Communications* 21 (1998) 243-253.
10. A. Gupta, D. O. Stahl and A. B. Whinston, The Economics of Network Management, *Communications of the ACM* 42(9) (1999) 57-63.
11. P. Reichl, D. Hausheer and B. Stiller, The Cumulus Pricing Model as an Adaptive Framework for Feasible, Efficient, and User-Friendly Tariffing of Internet Services, *Computer Networks* 43 (2003) 3-24.
12. M. A. Einhorn, Pricing and Competition Policies for the Internet, in *Public Access to the Internet*, eds. B. Kahin and J. Keller (Eaglewood Cliffs, NJ, Prentice Hall, 1994), pp. 338-349.
13. A. Gupta, D. O. Stahl and A. B. Whinston, Priority Pricing of the Integrated Services Networks, in *Internet*

- Economics, eds. L. W. McKnight and J. P. Bailey (3rd edn., Cambridge, MA, The MIT Press, 1997), pp. 323-352.
14. J. Altmann and K. Chu, How to charge for Network Services – Flat-Rate or Usage-Based, *Computer Networks* 36 (2001) 519-531.
 15. J. K. Mackie-Mason and H. Varian, Some FAQs about Usage-Based Pricing, *Computer Networks and ISDN Systems* 28 (1995) 257-265.
 16. C. Courcoubetis, F. Kelly and R. Weber, Measurement-Based Usage Charges in Communications Networks, *Operations Research* 48(4) (2000) 535-548.
 17. C. Courcoubetis, F. Kelly and R. Weber, Measurement-Based Usage Charges in Communications Networks, in *Technical Report-19* (Statistical Laboratory, University of Cambridge, Cambridge, UK, 1997).
 18. W. Pan, L. Yu, S. Wang and X. Wang, A fuzzy multi-objective model for provider selection in data communication services with different QoS levels, *International Journal of Production Economics* 147 (2014) 689-696.
 19. N. Kasap, H. H. Turan, H. Savran, B. Tektas-Sivrikaya and D. Delen, Provider selection and task allocation in telecommunications with QoS degradation policy, *Annals of Operations Research* (2016) DOI 10.1007/s10479-016-2213-5.
 20. H. H. Turan, N. Kasap and H. Savran, A bandwidth sourcing and task allocation model in telecommunications under stochastic QoS guarantees, *Applied Soft Computing* 24 (2014) 1112-1123.
 21. H. H. Turan, N. Kasap and H. Savran, A Hybrid Genetic Algorithm for Provider Selection and Task Allocation Problems in Telecommunications Networks, in *International Conference on Advanced Technology & Sciences* (Antalya, Turkey, August 12-15, 2014), pp. 610-615.
 22. D. T. Pham and D. Karaboga, *Intelligent Optimization Techniques* (Springer-Verlag, Great Britain, 2000).
 23. D. L. Olson and D. Delen, *Advanced data mining techniques* (Springer Science & Business Media, 2008).
 24. A. Oszyczka, *Evolutionary Algorithms for Single and Multicriteria Design Optimization* (Physica-Verlag, New York, 2002).
 25. F. T. S. Chan, K. C. Au, L. Y. Chan and T. L. Lau, Using genetic algorithms to solve quality-related bin packing problem, *Robotics and Computer Integrated Manuf.* 23 (2007) 71 – 81.
 26. K. J. Wang and Y. S. Lin, Resource allocation by genetic algorithm with fuzzy inference, *Expert Systems with Applications* 33 (2007) 1025 – 1035.
 27. H. Dawid, Adaptive Learning by Genetic Algorithms: Analytical Results and Applications to Economic Models, in *Lecture Notes in Economics and Mathematical Systems* (441, Springer-Verlag, Germany, 1996).
 28. M. Stillwell, D. Schanzenbach, F. Vivien and H. Casanova, Resource allocation algorithms for virtualized service hosting platforms, *Journal of Parallel Distributed Computing* 70(9) (2010) 962-974.
 29. G. Zhou, M. Hokey and M. Gen, A genetic algorithm approach to the bi-criteria allocation of customers to warehouses, *International Journal of Production Economics* 86 (2003) 35–45.
 30. P. Cortés, J. Larrañeta, L. Onieva, J. M. Gracia and M. S. Caraballo, Genetic algorithm for planning cable telecommunication networks, *Applied Soft Computing* 1 (2001) 21 – 33.
 31. J. K. Kim, S. Shivle, H. J. Siegel, A. A. Maciejewski, T. D. Braun, M. Schneider, S. Tideman, R. Chitta, R. B. Dilmaghani, R. Joshi, A. Kaul, A. Sharma, S. Sripada, P. Vangari and S. S. Yellampalli, Dynamically mapping tasks with priorities and multiple deadlines in a heterogeneous environment, *Journal of Parallel and Distributed Computing* 67 (2007) 154 – 169.
 32. F. Luna, A. J. Nebro, E. Alba and J. J. Durillo, Solving large-scale real-world telecommunication problems using a grid-based genetic algorithm, *Engineering Optimization* 40(11) (2008) 1067 – 1084.
 33. Z. H. Jin, K. Ohno, T. Ito and S. E. Elmaghraby, Scheduling Hybrid Flowshops Circuit Board Assembly Lines, *Production and Operations Management* 11 (2002) 216 – 230.
 34. S. Gul, B. T. Denton, J. W. Fowler and T. Huschka, Bi-Criteria Scheduling of Surgical Services for an Outpatient Procedure Center, *Production and Operations Management* 20(3) (2011) 406 – 417.
 35. S. Martello and D. Vigo, Exact Solution of the Two-Dimensional Finite Bin Packing Problem, *Management Science* 44(3) (1998) 388-399.
 36. K. J. Wang, S. M. Wang and S. J. Yang, A resource portfolio model for equipment investment and allocation of semiconductor industry, *European Journal of Operational Research* 179 (2007) 390 – 403.