Dynamic Multi-Dimensional PSO with Indirect Encoding for Proportional Fair Constrained Resource Allocation

Jonathan Hudson
University of Calgary
2500 University Dr. NW
Calgary, Canada
jwhudson@ucalgary.ca

Majid Ghaderi
University of Calgary
2500 University Dr. NW
Calgary, Canada
mghaderi@ucalgary.ca

Jörg Denzinger
University of Calgary
2500 University Dr. NW
Calgary, Canada
denzinge@ucalgary.ca

ABSTRACT

Dynamic particle swarm optimization (PSO) problems are generally characterized by the exhaustively examined issues of the changing location of optima, the changing fitness of optima, and measurement noise/errors. However, the challenging issue of continuously changing problem dimensionality has not been similarly examined. Given that in anytime dynamic resource allocation it is necessary to maintain a high quality solution, we argue that, rather than restarting the PSO algorithm, a more appropriate approach is to design an algorithm that robustly handles changing problem dimensionality. Specifically, we propose an indirect particle encoding scheme specifically designed for a dynamic multi-dimensional PSO algorithm for proportional fair constrained resource allocation. This PSO algorithm is implemented for the proportional fair allocation of power and users to channels within a simulation of an Orthogonal Frequency-Division Multiple Access (OFDMA) wireless network with mobile users switching cells as they traverse the simulation environment. The proposed PSO algorithm is evaluated using simulations, which demonstrate the ability of the proposed indirect encoding scheme to maximize the overall proportional fair optimization goal, without unfairly penalizing the individual components of the solution related to newly introduced problem dimensions.

Categories and Subject Descriptors

G.1.6 [**Optimization**]: Constrained optimization;
C.2.1 [**Network Architecture and Design**]: Wireless communication

General Terms

Algorithms, Performance, Experimentation

Keywords

Dynamic Particle Swarm Optimization, Multi-Dimensional, Constrained Resource Allocation, Proportional Fair, OFDMA

GECCO'14, July 12–16, Vancouver, BC, Canada.
Copyright 2014 ACM 978-1-4503-2662-9/14/07 ...$15.00.
http://dx.doi.org/10.1145/2576768.2598387.

1. INTRODUCTION

Real-world optimization problems are often dynamic in nature, due to alterations in the search space resultant from environmental or system changes. These changes require optimization to be continuous in some manner, through either restarting the optimization, or adapting the current optimization attempt on the fly. Depending on the application, restarting the optimization attempt is often undesirable due to the loss of discovered information and reduplication of effort. This is especially true, if the search space of the new problem is highly similar to that of the previous problem. However, the method of adapting the previous optimization attempt faces the challenge that the new optimization may remain stuck in a previously found optimum and therefore be unable to transition to find new local optima.

In particle swarm optimization (PSO) for dynamic optimization problems there are a number of sources of change for the search space. These include noise, location of optima, changes in fitness of optima, and changes in problem dimensionality. Noise in the problem is generally the result of stochastic influences on the fitness function, such as measurement or discretization errors [15]. Nickabadi et al. observed the following types of changes in optima: Type I as a change in optimum location, Type II as a change in optimum fitness with a fixed location, and Type III as a change of both optimum fitness and location [13]. These changes are subject to temporal severity (change occurs quickly) and/or spatial severity (local optima move a large distance in the search space as defined by the fitness function). These first three problems have been addressed in a wide variety of manners depending on the optimization problem's requirements, while the problem of dimensionality has not been explored to the same extent.

The problem of dimensionality changing has been examined by Kiranyaz et al. [9]. However, Kiranyaz et al. formulate the multi-dimensional problem with the range of dimensionality being a component of the search space. This means that the algorithm searches for both the optimal solution within a number of dimensions and which number of dimensions is optimal. However, multi-dimensionality is not always a component of the current working optimization search space. For example, the search space may be associated with a shared resource allocation problem for a certain number of agents. If new heterogeneous agents and their unknown constraints and relationship to current agents are entered into the search problem during optimization, then the dimensionality of the problem changes. More importantly, this possibility cannot be represented within the previous search space and therefore Kiranyaz et al.'s method does not apply.

The challenge addressed in this paper is to dynamically incorporate new search space dimensions by adapting the current PSO optimization attempt. This challenge is addressed for the problem of proportional fair allocation of shared resources in a cellular network with mobile users. The most common solution for the majority of dynamic search space changes is to maintain diversity within the PSO swarm's population such that the algorithm does not get trapped in local optima. However, in constrained proportional fair resource allocation there is a high dependency between the new dimensional values and current values, due to their relationship sharing a constrained resource (e.g. transmission power). This means there is high spatial severity and depending on the time requirements, temporal severity, in the impact of dimensional changes on the locations of local optima.

This paper develops a continuous dynamic PSO solution for the problem of changing dimensionality within optimizing the proportional fair distribution of shared and constrained resources in a cellular network employing OFDMA [18]. This solution takes the challenge of high dependency between new/existing values and through a unique indirect encoding method turns it into an advantage for robustly handling the changing dimensionality of the problem. More particularly, through the indirect encoding the proportional fair nature of the resource allocation problem is exploited to enable the incorporation of new dimensional values gracefully into the particle and fitness function. This inclusion is accomplished in a manner that the initial values of the new dimensions represent a relatively good start in regards to finding more optimal values. Importantly, the developed method allows for previous population and solution to be utilized in the subsequent optimization within the new search space, improving the quality and efficiency of the continuous optimization. Additionally, the PSO solution has the anytime property, quickly finding a valid, yet approximate and possibly suboptimal, solution and improving on it continuously for as long as required [26].

The rest of the paper is organized as follows. Section 2 examines the background of dynamic PSO. Then, Section 3 introduces the general method of continuous dynamic PSO optimization for the problem of proportional fair constrained resource allocation under the associated conditions of changing dimensionality. An application of this indirect encoding to continuous wireless OFDMA resource allocation with mobile users is outlined in Section 4. Numerical evaluations of the general method, through the application to OFDMA resource allocation, are completed in Section 5 demonstrating the ability of the method to robustly handle the changing dimensions of the search space. The evaluations demonstrate that the method is beneficial to the overall fitness of the problem and the individual components of the fitness related directly to the new dimensions. Finally, Section 6 concludes by examining the contributions of this paper and possible future work.

2. DYNAMIC PSO

Particle swarm optimization is a stochastic optimization method exploiting the concept of swarm intelligence [10]. The population for a PSO algorithm consists of a swarm of particles where each n-dimensional particle corresponds to a prospective solution within the current search space. In each iteration, every particle is updated using its velocity, its personal best position, and the swarm's global best position, along with a random influence. The swarm's global best encourages convergence towards a global optimum, the personal best influences convergence towards local optima, and the randomness acts as a stochastic influence. In each iteration of the algorithm, every particle is evaluated using the chosen fitness function in order to determine the particle's fitness value for its current location in the search space. If a better particle position in the search space has been found, then using this value the particle's personal and global best are updated.

Our PSO algorithm is designed to maintain a valid solution for a resource allocation problem at all times (anytime property [26]). To accomplish this, each dimension is limited strictly within the chosen valid boundary values. Consequently, the PSO consists of a swarm W of m particles P of n dimensions in length, and the swarm's global best $gbest = (gb_1, ..., gb_n)$ such that $W = (P_1, \ldots, P_m, gbest)$.

Each particle P is a 3-tuple, $P_i = (X_i, V_i, pbest_i)$, consisting of the particle's position $X_i = (x_1, ..., x_n)$, velocity $V_i = (v_1, ..., v_n)$, and personal best $pbest_i = (pb_1, ..., pb_n)$. The particle's dimensions are bounded as within a range with $x_j, pb_j, gb_j \in [0, 1]$, and velocity is bounded as $v_j \in [-0.1, 0.1]$.

The PSO algorithm begins with a swarm W consisting of particles with randomly initialized positions and velocities. For each next iteration $k+1$ from the current iteration k, each particle is updated according to the following equations:

$$V_i^{k+1} = w \cdot V_i^k + c_1 \cdot rand() \cdot (pb_i - X_i^k) + c_2 \cdot rand() \cdot (gb_i - X_i^k).$$

The velocity is changed based upon the previous velocity V_i^k and an inertial weight w, as well as the difference between the current position and the particle's personal best influenced by a learning factor c_1 and a random number in the range $[0, 1]$, and the difference between the current position and the particle's neighborhood best influenced by a learning factor c_2 and a random number in the range $[0, 1]$. This new velocity is limited such that each $v_j \in V_i^{k+1}$ is bounded by $v_j \in [-0.1, 0.1]$ with excess velocity beyond the bounds being discarded.

Next, the position is changed using the new velocity as described by the following equation:

$$X_i^{k+1} = X_i^k + V_i^{k+1}.$$

This new position is limited such that each $x_j \in X_i^{k+1}$ is bounded by $x_j \in [0, 1]$ with excess dimensional values beyond the bounds being discarded.

The basic PSO algorithm, as described by Eberhart et al., is unsuited for dynamic problems. Issues such as outdated memory as the search space changes (as stored in the global best of the swarm and personal best of the particles), lack of search space change detection, and the loss of diversity over time limit this base PSO algorithm to static problems [3]. Outdated memory is addressed in this paper using the common method of re-evaluating the fitness function of the personal and global best as the search space changes, such as in [8]. A common solution for search space change detection is sentry particles [6]. However, this is unnecessary in this paper as we know the problem changes quickly and continuously and therefore our solution requires no such trigger.

Diversity loss is a more complex challenge and the chosen technique for addressing it relies on the circumstances of the given problem. The general techniques as listed in [22] are: multi-population approaches, memory-based methods, and to increase population diversity after change. The first multi-population method is suited for multiple moving optima [14], while memory-based techniques are useful for problems with recurring search space states [23]. Our problem deals with a single optimum which is altered by a search space change with limited to no recurrence, and thus can be dealt with the random immigrants technique [20] which adds diversity (new particles) after a search space change.

One unaddressed area for PSO, which is explored in this paper, consists of dynamic optimization problems in which the dimensionality of the problem changes along with the other changes in the search space. Note, this is different from what was done by Kiranyaz et al. in [9] where it was possible to search within a known range of dimensionality of a specific search space. In the work of Kiranyaz et al. finding the optimal solution to a specific problem also consisted of simultaneously searching for an associated optimal number of dimensions. This paper deals with a dynamic search space in which the dimensionality of the problem will change indeterminately over time, resulting in unpredictable changes in the search space.

In the following section, an abstract continuous dynamic PSO solution to the problem of changing dimensionality is described. This solution tackles the challenge of a high dependency between existing and new dimensional values as the result of attempting proportional fair distribution of a constrained shared resource. The dynamic PSO solution is unique as it utilizes an indirect particle encoding method to incorporate new dimensions and their associated values within a particle into the existing solution. This is done in such a way that the new values are a relatively good starting point in regards to finding more optimal values, allowing for previously found populations and solutions to be utilized in the subsequent optimization.

3. PROPORTIONAL FAIR CONSTRAINED RESOURCE ALLOCATION

In this section, first the resource allocation problem and the associated assumptions necessary for the given solution are defined, then this paper's unique indirect encoding of this problem as a PSO particle is described.

3.1 Problem Definition

We are solving a constrained shared resource allocation problem with the assumptions that:

- We have a set $R = \{r_0, \ldots, r_{n_{res}-1}\}$ of n_{res} shared resources where each resource $r_j \in R$ has a constrained resource quantity $r_j^{quant} > 0$.
- We want to determine a proportionally fair allocation of the shared resources by determining a set $S = \{s_0, \ldots, s_{n_{sh}-1}\}$ of n_{sh} where each share $(port_i, r_j) = s_i \in S$ consists of a portion $port_i \in [0, r_j^{quant}]$ of a resource $r_j \in R$.
- The total of all portions $port_i$ related to the same resource r_j is equal to r_j^{quant}. That is, every resource represented by at least one share is allocated completely between all referencing shares.
- It is expected that at least two shares may reference (share) the same resource during optimization.
- Every resource does not need to be allocated to a share, but every share needs a resource even if the portion is assigned zero as an amount.
- Most importantly, the set S of shares is dynamic and the number of shares n_{sh} will increase and decrease over the continuous optimization process.

The optimization goal is the sharing of the resources R in a proportionally fair manner by determining assignments of the shares in S. It is important to note that this is proportionally fair in regards to the fitness function, which may value equal portions of a share differently depending on the entity/agent/system resource/etc. associated with that share. Therefore, a fair solution will not necessarily be the division of every resource equally between all the shares. For example, each share s_i has a logarithmic utility U of benefit, weighted by w_{r_j}, to the end user for utilizing the portion $port_i$ of resource r_j such that $U(port_i) = log(w_{r_j} \cdot port_i)$. The fitness function applied to a particle representing a resource allocation solution maximizes the summation of these utilities.

$$fit(X) = fit(S) = \sum_{s_i \in S} U(port_i)$$

In addition to the constrained OFDMA resource allocation problem considered in this paper [4,5], some examples of constrained resource allocations problems include: resource constrained multi-robot allocation which assigns robots to solve tasks given resource constraints [7], the allocation of constrained resources within a grid system to satisfy the performance requirements of the user's and system [12], and the constrained allocation of energy within data centers to maximize task completion [25].

3.2 PSO Indirect Encoding Solution

A solution to a constrained shared resource allocation problem consists of an assignment of resources from R to the shares in S. Each resource allocation solution is encoded as a particle in the PSO swarm in the following manner.

Every share $s_i \in S$ can be defined in one of two ways: with a known resource which does not need to be searched for during optimization, or an unknown resource that must be searched for during optimization. The most efficient method to encode a resource allocation solution as a particle in a PSO swarm is to encode a single dimension for every share s_i with a known resource (this dimension represents the portion $port_i$ of that resource), and two dimensions for every share with an unknown resource (these dimensions represent the resource r_j and the portion $port_i$ of that resource). However for simplicity of presentation in the following encoding definition, two dimensions x_{2i} and x_{2i+1}, and their accompanying velocities, will be encoded for each share s_i. A resource allocation solution

s_0	s_1	$\ldots$	$s_{n_{sh}-1}$

is expanded to

$r_{j'}$	$port_0$	$r_{j''}$	$port_1$	$\ldots$	$r_{j'''}$	$port_{n_{sh}-1}$

with $r_{j'}, r_{j''}, r_{j'''} \in R$ and is mappable one-to-one with

x_0	x_{0+1}	x_2	x_{2+1}	$\ldots$	$x_{2(n_{sh}-1))}$	$x_{2(n_{sh}-1)+1}$

To determine the resource referenced by a share with an unknown resource the particle value $x_{2i} \in X$ associated with the share's resource must be converted into a resource reference. The range $[0, 1]$ of the dimension of x_{2i} is consequently divided into equally sized segments for each resource r_j valid for the share s_i. The share's assigned resource is determined by which segment the value x_{2i} falls within. This is done using the following equation $r_j = r_{\lfloor x_{2i} \cdot |R^{s_i}| \rfloor}$ where $R^{s_i} \subseteq R$ consists of the valid resources in R for share s_i.

Now that the resources referenced by all shares are known, the portion $port_i$ for each share s_i can be determined by converting the value $x_{2i+1} \in X$ associated with the share's portion into a portion value $port_i$. We will consider the determined resource of a given share to be r_j. This portion is equal to the quantity of the resource r_j^{quant}, multiplied by the fraction of the current share's portion value x_{2i+1}, divided by the sum total of the portion values for every share sharing the current share's resource. That is

$$port_i = r_j^{quant} \cdot \frac{x_{2i+1}}{\sum_{r_j \equiv r_k} x_{2k+1}}$$

In each regular iteration of the PSO, all particles in the swarm will be updated according to the standard PSO algorithm defined in the previous section. In order to evaluate the fitness function, each particle is then converted via this encoding into a resource allocation solution and the fitness function is used to determine the value of the particle's location in the search space. The optimization process is never reset, rather diversity is maintained by adding new random immigrants periodically.

If a new share is created, n_{sh} is incremented and a new share $s_i = s_{n_{sh}-1}$ is added to S and the dimensions, x_{2i} and x_{2i+1} required to represent it, are added to every particle/velocity in the swarm, including the global/personal best, with new random values. If a share s_i is removed, then n_{sh} is decremented and the dimensions, x_{2i} and x_{2i+1} relating to the share, are similarly removed. At either of these times the fitness values of the particles and global/personal best are re-evaluated.

The chosen indirect encoding ensures that when dimensions for a new share are added, then the share is already referencing some resource r_j and is being given a portion $port_i$ of it. By determining this portion by utilizing a fraction of the summation of portions referencing the same resource, this initial portion is taken proportionally from all of the other shares. This may not exactly be the optimal proportionally fair solution, especially in regards to the utility function U, but it is often a good initial value such that the agent/entity/etc. associated with the new share is not unfairly penalized by being new to the problem.

To demonstrate these claims, the following section implements this particle encoding method for the problem of OFDMA cellular bandwidth allocation with mobile users entering and leaving the service of cell towers.

4. OFDMA CELLULAR BANDWIDTH ALLOCATION

In order to validate the described continuous dynamic PSO for proportional fair multi-dimensional search, this paper applies it to the problem of proportional fair Orthogonal Frequency-Division Multiple Access (OFDMA) resource allocation. Many high-rate wireless data services, such as the 4G cellular services standard Long Term Evolution (LTE), employ OFDMA to allocate radio resources to users, based on channel condition, in order to improve the overall performance of the cellular system.

The dynamic multi-user OFDMA literature generally divides optimization techniques into two classes: margin adaptive (MA) [2] and rate adaptive (RA) [17] techniques. MA schemes attempt to minimize overall transmit power, while RA schemes attempt to maximize the total data rate. Of particular interest for this paper are existing RA applications of well performing heuristic techniques such as simulated annealing [4], game theoretic [11], genetic algorithm [21], and particle swarm optimization (PSO) [2, 16, 17, 24] approaches. Unlike MA algorithms, which have free capacity for users, RA algorithms have to redistribute resources (i.e. power, time) away from other users to handle an increase in load resulting in a high dependency between users' resource allocations.

PSO techniques have been applied rather successfully to the problem. However, the papers utilizing RA PSO techniques do so in a static setting, which does not address the issue of solving the continuous dynamic optimization problem caused by user mobility [17,24]. As users leave/enter the coverage of a cell, so do the user's accompanying resource allocation requirements. Therefore, it is desirable for the network to gracefully incorporate the additional load of new mobile users while maintaining appropriate data rates for existing users. Additionally, it is important that the system maintains an appropriate valid solution such that users are always provided cellular service (i.e. the anytime property).

4.1 OFDMA Model

A static version of the dynamic rate adaptive OFDMA model used in this paper, which ignores the difficulty of user mobility and hand-off between cells, can be found in [4]. In our dynamic OFDMA model, let K denote the set of separately managed wireless cells. In this paper, each individual cell's resource allocation problem is optimized separately. Let, M and N denote, respectively, the set of users and channels (sub-carriers) each cell has available. Each user $i \in M$ has a proportional fair throughput utility function $U(Rate_i) = \log(Rate_i)$. Proportional fairness among users is then achieved by maximizing the summation of the logarithm of individual users' throughputs $\sum_{i \in M} U_i(Rate_i)$.

In order to determine this rate, the resource of transmission power at each cell is allocated to available cell channels and the users are allocated each to a single channel for a portion of time. First, each cell $k \in K$ allocates a portion of power P_{jk} to channel $j \in N$ under its maximum total transmit power p_k^{max} such that $\sum_{j \in N} P_{jk} \leq p_k^{max}, \forall k \in K$. Second, $\tau_{ijk} \geq 0$ is the time portion granted by a cell to user $i \in M$ on channel $j \in N$ for cell $k \in K$ with $\sum_{i \in M} \tau_{ijk} \leq 1, \forall j \in N, \forall k \in K$.

The total throughput $Rate_i$ that is achieved by user $i \in M$ is therefore

$$Rate_i = \sum_{j \in N} \sum_{k \in K} q_{ik} \cdot \tau_{ijk} \cdot C(S_{ijk}).$$

The variable $q_{ik} \in \{0, 1\}$ indicates if a user $i \in M$ is assigned to a cell $k \in K$ where each user is served by a single cell such that $\sum_{k \in K} q_{ik} = 1, \forall i \in M$. The Shannon-Hartley capacity formula $C(S_{ijk}) = \log(1+S_{ijk})$ is used to determine the feasible transmission rate based on the signal to noise ratio (SINR) S_{ijk} of user $i \in M$ served by cell $k \in K$ on channel $j \in M$ [19]. The Signal-to-Interference-plus-Noise-Ratio (SINR) of user $i \in M$ being served by cell $k \in K$ on channel $j \in M$ is $S_{ijk} = \frac{G_{ik}P_{jk}}{\eta + \sum_{l \neq k} G_{il}P_{jl}}$, with η being the thermal background noise power. Additionally, G_{ik} is the power gain of user $i \in M$ for cell $k \in K$, with each cell having a strict finite communication range resulting in a gain of zero if the user is outside the cell's range.

4.2 PSO for OFDMA

OFMDA resource allocation has typically been approached as a static problem, relying on the optimization to be reset when the problem search space changes. OFDMA resource allocation is a continuous problem, with the changing environment conditions and mobile user demands resulting in both the dimensions and the search space of the problem to be in flux. The following three problems must be dealt with for proportional fair OFDMA dynamic resource allocation: (1) the algorithm should not get stuck in local optima as the search space changes, (2) it is undesirable to restart the search every time a user leaves/enters service, and (3) it is undesirable that users are disproportionally penalized when they change between serving cells.

The first of these problems will be addressed by maintaining diversity through regularly introducing random immigrants into the swarm's population [20] to avoid unwanted homogeneity and retain flexibility during the search. The second problem is addressed through the expandable multi-dimensional nature of the particle in this paper's PSO. The particle expands with two new dimensions for every new user, and contracts similarly by deleting the two dimensions referenced by a leaving user. Every time this occurs the old personal and global bests, are re-evaluated, as in [8].

The final problem is addressed by the interpretation of the particle values as shares in the resource allocation solution. An arriving user is assigned two new dimensions with random values. Given the interpretation method, this portion of time on the assigned channel is taken equally from each of the other users already assigned to the channel. This allows for the transitioning user to receive an initial acceptable level of service, without requiring the other users served by the cell to sacrifice their own service unfairly. Similarly, a leaving user's time is redistributed to the remaining users. From this state, the search algorithm will continually optimize towards possible better solutions, shifting users among channels, changing channel power levels, and user time portions.

The remaining parts of this subsection describe the formal application of the abstract PSO solution from the previous section to the problem of dynamic constrained OFDMA resource allocation with mobile users. First, an OFDMA resource allocation particle encoding is described as an implementation of the general resource allocation encoding. Then, an example is given to demonstrate a particle's interpretation.

4.2.1 OFDMA Encoding

Our resources R consist of a resource of transmission power and a set of channels $|N|$. Our set of shares S consists of a share for every channel that the power (known resource) will be split between, and a share for every user that requires service from the cell on some channel (unknown resource). The OFDMA encoding for a single cell $k \in K$ has to include three parts: (1) the fraction of the total power P^k_{max} assigned to each channel $j \in N$, (2) the channel resource for each served user, and (3) the time portion for each served user on his assigned channel. Note, a cell k serves a set of users O where $O \subseteq M$ is a subset of all users.

An n-dimensional vector therefore consists of two portions: (1) a static portion of $|N|$ shares which maps to assignments of the known power resource to a channel $j \in N$, and (2) a dynamic portion of $2 \cdot |O|$ shares which maps two vector values for each user to a portion of time on a single channel resource. Therefore, the size of a vector is $n = |N| + 2 \cdot |O|$, which dynamically changes as users are added and removed from a cell. The vector encoding is defined

pow_0	...	$pow_{\|N\|-1}$	$user_0$	...	$user_{\|O\|-1}$

with the unknown resource user $i \in O$ shares defined

$$user_i = \boxed{chan_i \mid time_i}$$

This encoding can be interpreted as an OFDMA resource allocation solution. First, the power p_{jk} for each channel $j \in N$ is determined via

$$p_{jk} = P^k_{max} \cdot \frac{pow_j}{\sum_{l \in N} pow_l}.$$

That is the power for each channel is found by totaling the power values in the vector and determining the portion of that total which consists of the channel's power value. This portion is multiplied by the total available power P^k_{max} to get the power assigned to the channel. Second, the channel $j \in N$ of the user $i \in O$ is $\lfloor chan_i \cdot |N| \rfloor$.

The range $[0, 1]$ is divided into an equal portion for each channel, and which portion the vector's user channel value falls within is the user's assigned channel. Finally, the time fraction of the user $i \in O$ on channel $j \in N$ is

$$\tau_{ijk} = \frac{time_i}{\sum_{chan_j \equiv chan_l} time_l}.$$

That is the time for the user on his assigned channel is found by totaling the time values in the vector for all users on the same channel. Then the portion of that total, given by the individual's time value, is determined. This portion is the time allocated to the user on his assigned channel.

4.2.2 Example

Consider the following example of a vector:

pow_0	pow_1	$user_0$		$user_1$		$user_2$	
0.25	0.25	0.33	0.45	0.66	0.30	0.70	0.20

For this example there are two channels, i.e. $|N| = 2$, and there are three users, i.e. $|O| = 3$. The total possible power in each cell is 16 W.

The following is how to interpret the encoding. First, the power p_{0k} for channel 0 is $p_{0k} = 16 \cdot \frac{pow_0}{\sum_{l \in N} pow_l} = 16 \cdot \frac{0.25}{0.5} = 8$. As well, the power p_{1k} for channel 1 is $p_{1k} = p_{0k} = 8$. Second, the channel of the user 0 is channel $\lfloor chan_0 \cdot 2 \rfloor = \lfloor 0.33 \cdot 2 \rfloor = 0$. The channel of the user 1 is channel $\lfloor chan_1 \cdot 2 \rfloor = \lfloor 0.66 \cdot 2 \rfloor = 1$. The channel of the user 2 is channel $\lfloor chan_2 \cdot 2 \rfloor = \lfloor 0.7 \cdot 2 \rfloor = 1$. Finally, the time fraction of the user 0 on channel 0 is $\tau_{00k} = \frac{time_0}{\sum_{chan_0 == chan_l} time_l} = \frac{0.45}{0.45} = 1$. The time fraction of the user 1 on channel 1 is $\tau_{11k} = \frac{time_1}{\sum_{chan_1 == chan_l} time_l} = \frac{0.3}{0.5} = 0.6$. The time fraction of the user 2 on channel 1 is $\tau_{21k} = \frac{time_2}{\sum_{chan_2 == chan_l} time_l} = \frac{0.2}{0.5} = 0.4$.

5. EXPERIMENTS

In this section, the evaluations completed to examine the OFDMA implementation of the PSO for proportional fair constrained resource allocation with changing dimensions are presented. It is important to mention that we know of no other **dynamic** OFDMA resource allocation methods to compare our method to. This is primarily because the existing developments for OFDMA resource allocation have focused on the static version of the problem. Therefore, to use one of these static techniques for the dynamic variant would require additional research to create a new dynamic development of a given static technique. Additionally, it would be an unproductive evaluation to compare the initial solutions of static techniques, which are made suboptimal and unsuitable as time passes and users move between cells, to the dynamic and continuously relevant solution created by our dynamic technique. First, Subsection 5.1 outlines the parameters of the OFDMA wireless model and the experimental configurations. Then, Subsection 5.2 describes the results of evaluating the performance of our indirect vector encoding for dynamic OFDMA resource allocation

5.1 Parameters

In this subsection, the parameters of the simulation environment, PSO algorithm, and wireless OFDMA model are described as used in the experimental evaluations.

5.1.1 Environment Parameters

The environment for the experiments is constructed similar to the environment in [4]. There are $|K| = 4$ cell sites located in a rectangular coverage area of $X_0 = 4$ km by $Y_0 = 4$ km. Each cell has a circular coverage diameter of 2.5 km. The edges of the environment are connected in a toroidal 'wrap-around' manner to avoid boundary effects. The cell sites are arranged in a diamond pattern in the environment to maximize coverage. The model is tested with three amounts of $|M| = 64$, 128, or 256 users, which are distributed at random across the coverage area. These users are each given a random static directional vector of (x, y) environmental movement velocity with a speed of at most 100 km/h. Additionally, the model is tested with three different counts of channels $|N| = 16$, 128, or 1024.

5.1.2 PSO Parameters

A cell is assigned a swarm of 25 particles, which every 100 iterations has 5 particles replaced by new random immigrant particles to maintain diversity. The learning factors are $c_1 = 1.0$ and $c_2 = 1.0$. At the beginning of an evaluation, users are randomly assigned to any cell which currently covers them for service. When a user leaves a coverage area, his associated particle dimensions are removed from the cell's swarm they are leaving and two new random dimensions are added and assigned to the user in the swarm of the cell they are arriving at. Each PSO swarm operates continuously and independently in its own thread. Similarly, the users' positions are updated independently by a thread which evaluates their current location every 0.1 seconds. In order to facilitate comparison between varying numbers of users, the fitness function (aggregate throughput utility) for the PSO is normalized by the number of users, obtaining the average throughput utility per user: $\frac{1}{|M|}\sum_{i=1}^{M}\log(Rate_i)$.

5.1.3 Wireless Parameters

The assumptions concerning channel gains are broadly consistent with the standard 3GPP propagation models [1]. In particular, the channel gain value from cell site k to user i is $G_{ik} = H(D_{ik})$, with D_{ik} the distance between cell site k and user i (in km), and $H(d) = 10^{h_0} \cdot d^{-\kappa}$, i.e. $H(d) = 10 \cdot h_0 - 10 \cdot \kappa \cdot log_{10}(d)$ (dB), with a path loss exponent $\kappa = 3.5$ and $h_0 = -14.4$. The thermal background noise is $\eta = -174$ dBm (Hz^{-1}), and the bandwidth per frequency is 1 MHz. Each cell site has a maximum total transmit power budget of $P_k^{max} = 16$ W. The feasible transmission rate as a function of SINR s is given by $C(s) = c_0 \cdot \log_2(1+s)$ (Kbps), with $c_0 = 1000$.

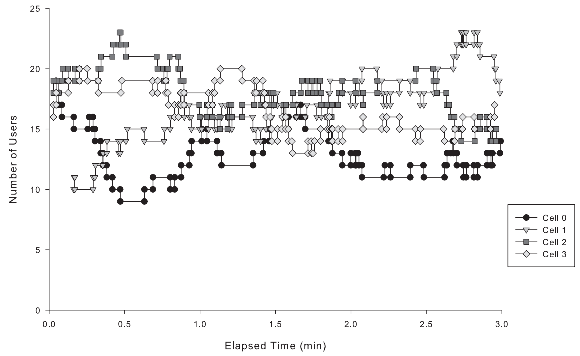
5.2 Numerical Evaluations

In order to complete the evaluations a simulator was developed. Four cells in a diamond pattern with overlapping coverage areas were simulated. In total, nine evaluations of 3 minutes in length were completed, each consisting of a unique combination of the user and channel counts stated in the previous subsection. We examine the results, first to see if the overall throughput of cell service was maintained, and second to see how the throughput of individual users was maintained as mobile users transit the environment and shift between cells. Finally, we examine the impact of changing the PSO parameters that maintain diversity and how the algorithm performs at different rates of user mobility.

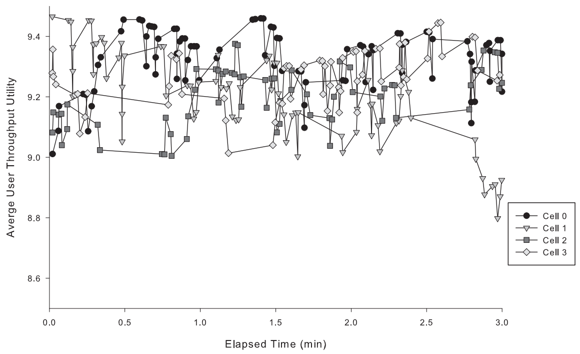
Figure 1 examines the overall service by reporting the average user throughput utility and served users for the 4 cells with 64 users and 16 channels. The other results for the remaining combinations are omitted as they show the same behavior with a slightly higher or lower convergence of average user throughput utility. Summarizing across all nine evaluations, an increase in channels available to a cell in general increases the ability of the cell to share and utilize available bandwidth, resulting in, on average, higher user throughput utility. This convergence is higher the more channels and less users being served per cell. Additionally, cross-referencing (a) and (b) in Figure 1 shows that the average user throughput utility of a cell is inversely correlated to the number of users being served by the cell.

Figures 2 and 3, of low and then high user load per channel, report the user throughput utility and serving cell for 4 selected users which switch between the 4 cells during the evaluation period. Each time a user transfers between cells the transfer is indicated by an additional point on the horizontal Elapsed Time axis. The graphs are combinations of 64/256 total users and 16/1024 channels. The other results are omitted as they do not provide as much detail as the chosen examples of more extreme combinations of low and high user per channel load.

In the case of low load usage the existence of unused channels, as in Figure 2, means that the initial random channel assignment for a user on a particular cell may be completely unpowered. This unpowered condition can be seen in User 0 and 1 who each transfer between cells and are assigned to an initial random unpowered channel and resultantly have no throughput. However, when this event happens the algorithm immediately rectifies the lack of power in the next update. This boundary case may be rectified easily in future algorithm development by limiting initial random channel assignment for new users to only channels with power. This is not a problem with high load usage, such as in Figure 3,



(a) Number of Users Served by each Cell



(b) Average User Throughput Utility for each Cell

Figure 1: Cell Service (64 Users, 16 Channels)

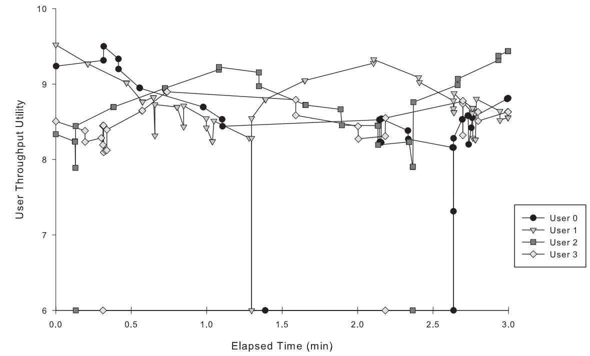


Figure 2: Low Load - 1 User per 16 Channels (64 Users, 1024 Channels)

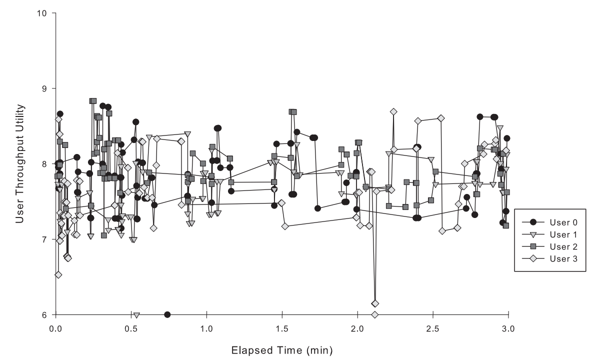


Figure 3: High Load - 16 Users per 1 Channel (256 Users, 16 Channels)

when there are more users than channels, as all channels are always powered.

In the case of high load usage, such as Figure 3, there is much more volatility to an individual user's transfer rate. This is because in most cases a user is sharing a channel with other users. Therefore, as the users' channel conditions change as they move around the environment, when the algorithm attempts to change the user's time portion on a channel it not only affects the user's own throughput but that of the other users on the same channel. A user transferring between cells can also experience additional volatility when the user's new time portion assignment on a channel is much lower than those of the other users sharing the channel. This is the case for user 1 in Figure 3. This may be rectified in future algorithm development by caching an average user portion for each channel which is assigned to newly arriving users instead of a completely random value. Volatility is not a problem with low load usage, such as in Figure 2, where after a user moves to a new cell their throughput is more stable, increasing as the user moves closer towards the cell center then decreasing as the user subsequently moves away.

Figure 4 examines the impact of changing the PSO diversity parameters. In the previous evaluations these parameters replaced 5 random particles in the PSO every 100 iterations to maintain the diversity of the swarm. This diversity is desirable as it means the PSO can avoid being unable to find the desired solution to the current problem because it has become too homogenous and is trapped in past local optima. In these evaluations the impact of variable rates of diversity change on the average throughput utility of a cell is reported for a single cell of the 4 for 64 users on 16 channels. The different chosen rates include never updating the diversity, or replacing a single particle every $10, 100, 1000$, and 10000 iterations of swarm. All the different rates face very short term drops in the average throughput rate, which is the expected result of the changing problem. However, never updating, or updating too rarely, result in longer periods around the 2 minute mark where the algorithm remains stuck in an old local optima and is unable to transition to the better solution found by the more diverse algorithms.

In summary, outside of the identified rectifiable edge cases, the algorithm maintains good initial rates for users switching between cells and the service level of existing users as new users arrive. The identified issues, with low channel load users being assigned to an unpowered channel and with high channel load users getting inadequate initial time portions, should be solvable with the described additions to the algorithm's behavior. The algorithm's diversity is an important factor in allowing the algorithm to provide relevant solutions to even challenging problems with highly mobile users.

6. CONCLUSION

This paper introduced a proportional fair constrained resource allocation PSO algorithm for dynamic multi-dimensional optimization problems and implemented it for rate adaptive OFDMA resource allocation with mobile users. This imple-

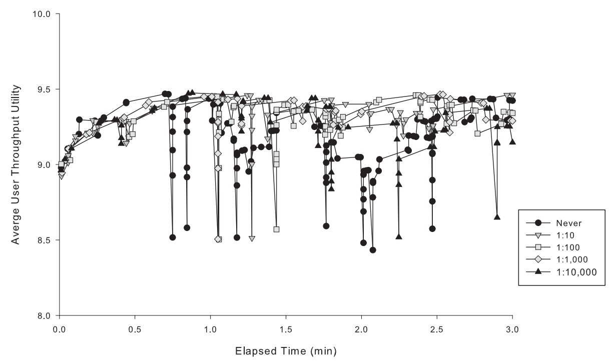


Figure 4: Diversity Maintenance (64 Users, 16 Channels)

mentation demonstrated, over a continuous period of operation, the ability to gracefully optimize the proportional fair allocation of resources within a cell as user load moved between cells. This load management optimization benefited both the overall cellular system's performance, as well as the data rates experienced by individual cellular users switching between cells. The implementation demonstrated the ability of the PSO algorithm to allocate a constrained and shared set of resources in a proportional fair manner for a dynamic problem with quickly changing dimensionality. However, the evaluations also indicated addressable boundary case weaknesses. First, when low load resulted in initial assignments of users to unpowered channels. Second, when high load resulted in volatility due to inadequate initial time portion assignment on shared channels. In the future, providing more considerations on the initial resource portion, such as a cached average instead of random value, may provide the ability to suppress this volatility.

7. REFERENCES

[1] "3GPP." Internet: *www.3gpp.org*, [Jan. 29, 2013].

[2] I. Ahmed and S. Majumder. Adaptive Resource Allocation Based on Modified Genetic Algorithm and Particle Swarm Optimization for Multiuser OFDM Systems. In *Proc. ICECE*, 211–216, 2008.

[3] T. Blackwell. *Particle Swarm Optimization in Dynamic Environments*. Springer, 2007.

[4] S. Borst, M. Markakis, and I. Saniee. Distributed Power Allocation and User Assignment in OFDMA Cellular Networks. In *Proc. Allerton*, 1055–1063, 2011.

[5] F. Brah, L. Vandendorpe, and J. Louveaux. OFDMA Constrained Resource Allocation with imperfect Channel Knowledge. In *Proc. COM/VT*, 1–5, 2007.

[6] A. Carlisle and G. Dozler. Tracking Changing Extrema with Adaptive Particle Swarm Optimizer. In *Proc. WAC*, vol.13, 265–270, 2002.

[7] Chen et. al. Resource Constrained Multirobot Task Allocation with A Leader-Follower Coalition Method . In *Proc. IROS*, 5093–5098, 2010.

[8] X. Hu and R. Eberhart. Adaptive Particle Swarm Optimization: Detection and Response to Dynamic Systems. In *Proc. CEC*, vol.2, 1666–1670, 2002.

[9] T. Ince, S. Kiranyaz, and M. Gabbouj. A Generic and Robust System for Automated Patient-Specific Classification of Electrocardiogram Signals. *IEEE Trans. Bio-Med. Eng.*, 56(5):1415–1426, May 2009.

[10] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *Proc. ICNN*, 1995, pp. 1942–1948.

[11] Z. Liang, Y. Chew, and C. Ko. Decentralized Bit, Subcarrier and Power Allocation with Interference Avoidance in Multicell OFDMA Systems using Game Theoretic Approach. In *Proc. MILCOM*, 1–7, 2008.

[12] Y. Liu, K. Han, and D. Zhang. Consumer Sharing Policy Constrained Resource Allocation Method for Grid System . In *Proc. ISCIT*, 721–725, 2009.

[13] A. Nickabadi, M. Ebadzadeh, and R. Safabakhsh. DNPSO: A Dynamic Niching Particle Swarm Optimizer for Multi-Modal Optimization. In *Proc. CEC*, 26–32, 2008.

[14] D. Parrott and X. Li. A Particle Swarm Model for Tracking Multiple Peaks in a Dynamic Environment using Speciation. In *Proc. CEC*, vol.1, 98–103, 2004.

[15] K. Parsopoulos and M. Vrahatis. Recent approaches to global optimization problems through Particle Swarm Optimization. *Natural Computing*, 1:235–306, 2002.

[16] S. Sadeque, I. Ahmed, and R. Vaughan. Impact of Individual and Joint Optimizations in Multi-user OFDM Resource Allocation by Modified PSO. In *Proc. CCECE*, 1233–1237, 2011.

[17] Sharma et. al. On the use of particle swarm optimization for adaptive resource allocation in orthogonal frequency division multiple access systems with proportional rate constraints. *Info. Sci.*, 182(1):115–124, 2012.

[18] T. Starr, J.M. Cioffi, P.J. Silverman. *Understanding Digital Subscriber Line Technology*. Prentice Hall PTR, 1999.

[19] D. Tse, P. Viswanath. *Fundamentals of Wireless Communications*. Cambridge University Press, 2004.

[20] H. Wang, D. Wang, and S. Yang. Triggered Memory-Based Swarm Optimization in Dynamic Environments. In *Proc. the EvoCOP*, 637–646, 2007.

[21] Y. Wang, F. Chen, and G. Wei. Adaptive Subcarrier and Bit Allocation for Multiuser OFDM System Based on Genetic Algorithm. In *Proc. ICC*, vol.1, 242–246, 2005.

[22] S. Yang. Population-Based Incremental Learning with Memory Scheme for Changing Environments. In *Proc. the GECCO*, 711–718, 2005.

[23] S. Yang and X. Yao. Population-Based Incremental Learning With Associative Memory for Dynamic Environments. *IEEE Trans. Evol. Comput.*, 542–561, October 2008.

[24] Y. Yi, Z. Yu, and W. Ye. Modified Particle Swarm Optimization and Genetic Algorithm Based Adaptive Resources Allocation Algorithm for Multiuser Orthogonal Frequency Division Multiplexing System. *Information Technology*, 10(5):955–964, 2011.

[25] Young et. al. Energy-Constrained Dynamic Resource Allocation in a Heterogeneous Computing Environment. In *Proc. ICPPW*, 298–307, 2011.

[26] S. Zilberstein and S. Russell. Approximate Reasoning Using Anytime Algorithms. In S. Natarajan, editor, *Imprecise and Approximate Computation*, 43–62. Springer US, 1995.