Evolutionary algorithm for analyzing higher degree research student recruitment and completion

Ruhul Sarker and Saber Elsayed

Cogent Engineering (2015), 2: 1063760
Evolutionary algorithm for analyzing higher degree research student recruitment and completion

Ruhul Sarker* and Saber Elsayed

Abstract: In this paper, we consider a decision problem arising from higher degree research student recruitment process in a university environment. The problem is to recruit a number of research students by maximizing the sum of a performance index satisfying a number of constraints, such as supervision capacity and resource limitation. The problem is dynamic in nature as the number of eligible applicants, the supervision capacity, completion time, funding for scholarships, and other resources vary from period to period and they are difficult to predict in advance. In this research, we have developed a mathematical model to represent this dynamic decision problem and adopted an evolutionary algorithm-based approach to solve the problem. We have demonstrated how the recruitment decision can be made with a defined objective and how the model can be used for long-run planning for improvement of higher degree research program.

Subjects: Evolutionary Computing; Intelligent Systems; Operations Research

Keywords: higher degree research student recruitment; differential evolution; evolutionary algorithms

ABOUT THE AUTHORS
Ruhul Sarker and Saber Elsayed are with the School of Engineering and Information Technology at University of New South Wales, Canberra, Australia. Ruhul Sarker’s research interests are on evolutionary computation, optimization and their interfaces. Saber Elsayed’s research interests include design of evolutionary computation-based algorithms and optimization. Jointly they have developed a number of high performing evolutionary algorithms for solving different optimization problems. Their contributions have been recognized through publications in high impact journals in the field such as IEEE Transactions on Evolutionary Computation, IEEE Transactions on Industrial Informatics, and Computers & Operations Research. The research reported in this paper includes a new dynamic optimization problem from a higher education institution and a solution approach for solving such a complex problem. There are many situations in practice where similar problems can be found. So the solution approach developed in this research will help to solve those problems.

PUBLIC INTEREST STATEMENT
The authors are well-known from their contributions in the field of evolutionary computation and optimization. They have applied their developed algorithms in solving many different practical problems. In this paper, they have introduced an interesting practical problem and solved it using an evolutionary algorithm. Practical problems from different domains, with similar characteristics, can be solved using the algorithms applied in this paper. This would benefit many organizations and community.
1. Introduction

Many real-world decision problems, such as project scheduling, production planning, and resource allocation, are multi-period and dynamic (Sarker & Newton, 2007). In these problems, it is required to make the decisions for many periods in the future repeatedly. In some cases, firstly, the multi-period problem is solved as a static problem based on the anticipated parameters and then the solutions are updated at a regular interval with the availability of new information. In some other cases, the problem is solved only for the current period as the data and information for future periods are hardly available in the current period. In the literature, similar problems are solved as optimization problems.

In this paper, we introduce a problem of higher degree research student recruitment (HDRSR) process. In the process, we recruit the best possible set of students considering the eligibility of students, their ability for timely completion, supervision capacity, funding availability, and other resource limitation. As all these information are available at the time of the application process, the decision problem is to allocate the eligible students to some supervisors or a group of supervisors in different disciplines using a department, school or faculty specific decision criteria. Examples of decision criteria are maximizing the throughput, minimizing the average completion time, maximizing the quality of output, minimizing the overall cost, and maximizing the return on investment. The HDRSR problem looks like a single-period static optimization problem that must be solved in each period (or session or semester) with period dependent parameters and data.

To analyze the recruitment pattern or the performance over a number of future periods, it is required to solve the problem for many periods, where the parameters are either stochastic or dynamic. That means, this is a dynamic optimization problem where the timing of parameter change is known, but their magnitude must be either calculated or predicted using historical data and derived functions. For better results, such a model must be run in each period on a rolling horizon basis.

In this research, we have defined a HDRSR as an optimization problem and developed a mathematical model to represent the problem. The mathematical model has been solved using a differential evolution (DE) algorithm. The representative data were taken from a research intensive school/department considering five research focus areas. The model has been solved using DE algorithm for a single semester as a static optimization problem and for several semesters in the future as a dynamic optimization problem. The performance of each research focus area has been analyzed and their combined affect has also been reported.

The rest of this paper is organized as follows: Section 2 presents the problem definition and mathematical model of HDRSR. Section 3 gives an overview of DE. The details of the algorithm used to solve the problem are given in Section 4, while Section 5 presents the computational results. Finally, conclusions are elaborated in Section 6.

2. Problem definition and mathematical modeling

In this paper, we consider an academic unit, such as a department, school or research centre, within a higher degree institution/university. As a part of the academic activities, the selected academic unit offers higher degree research program, such as Doctor of Philosophy (PhD). The unit has a number of research focus areas and each area has limited qualified academics to supervise HDR students. The unit has its own scheme to fund scholarships to high quality HDR students. There may be some additional funding available under special projects and from external sources. The eligible students apply for admission to conduct research in a research area of his/her interest and at the same they apply for scholarships. In the HDRSR problem, for simplicity, we aim to allocate the research students to a group of supervisors, having similar research interests, by maximizing the sum of a performance index that emphasizes on quick completion as well as high quality output. However, one can consider the allocation to individual supervisors or school and faculty level. Such a problem requires input from the academic unit, such as supervision capacity in each research focus area in each semester, the number of scholarships available in each semester, the number of eligible applicants under each area
in each semester, and any special condition imposed by the unit, such as supervision performance of each research area. The output expected is the allocation of students to each research area and performance analysis for each area and the unit as a whole.

We assume that all applicants eligible for enrollment will apply for scholarships. Here, the scholarship means living allowance paid by the individual research group, or the academic unit. This assumption can be relaxed depending on the mix of applicants. For example, a group of students may require only supervision not funding. We consider two types of scholarships: (1) common pool—applicants from any research group can apply and (2) special scholarships—funded by individual research group for their own applicants. We also assume that the expected completion time for a PhD program is 3.5 years (seven semesters).

The constraints considered here are the number of eligible applicants, the supervision capacity in each research area, and the availability of funding for scholarships (converted to the number of scholarships) in both general pool and special scholarships.

To develop the mathematical model for the HDRSR problem, we define a set of decision variables and parameters, as presented in Table 1.

The objective is to maximize the sum of a performance index subject to a set of constraints, such that

\[
\text{Maximize } f(x) = \sum_{i}^{n} \left( \frac{W}{ACT_{it} - D} \right) x_{it}
\]

Subject to:

1. \[x_{it} \leq NA_{it}, \quad \forall i, t\]  
2. \[x_{it} \leq SC_{it}, \quad \forall i, t\]  
3. \[y_{it} \leq SS_{it}, \quad \forall i, t\]  

### Table 1. List of abbreviations used

| Index | Meaning |
|-------|---------|
| i     | A research group |
| t     | Time period (equal to a semester) |
| k     | An applicant (eligible research student) |
| \(x_{it}\) | Number of students recruited to start in \(t\) with \(i\) |
| \(y_{it}\) | Number of students selected for special scholarships in \(t\) with \(i\) |
| \(NS_{it}\) | Number of scholarships available in the general pool in \(t\) |
| \(SS_{it}\) | Number of special scholarships available in \(i\) at semester \(t\) |
| \(NA_{it}\) | Number of eligible applicants in \(i\) at \(t\) |
| \(SC_{it}\) | Supervision capacity available (number of students that can be supervised) in \(i\) at \(t\) |
| \(SC_{\text{com}}_{it}\) | Number of students successfully completed their degree from \(i\) at \(t\) |
| \(NPub_{it}\) | Number of elite papers published from student research in group \(i\) in semester \(t\) |
| \(DES_{kit}\) | Number of semesters taken to complete by a student \(k\) from \(i\) (who completes in semester \(t\)) |
| \(ACT_{it}\) | Average completion time of students of group \(i\) as calculated in semester \(t\) |
| \(D\) | The minimum number of semesters allowed by the university for thesis submission−1 |
| \(QI_{it}\) | Quality index of students’ research in group \(i\) in semester \(t\) |
| \(n\) | The number of groups |
| \(W\) | The weight |
where the constraint inequalities Equations 2–6 represent the limit on eligible applicants, supervision capacity, limit on special scholarships, limit on general pool of scholarships, and non-negativity of variables, respectively. Firstly, the applicants are ranked based on their qualifications, research experiences, and publications, and then the number of short-listed applicants set the limit on eligible applicants used in the constraint. The supervision capacity is determined based on the number of students that can be supervised by the research area in a semester. The number of scholarships is calculated based on the funding available. The average completion time is calculated, and updated at each \( t \), using the following equation that considers the completions of the past \( S \) semesters

\[
ACT_{it} = \frac{\sum_{t-S}^{t-1} \sum_{k} DES_{kit}}{\sum_{t-S}^{t-1} SCom_{it}}
\]  

For simplicity, in this paper, the quality index is measured as the high quality publications produced per unit supervision capacity, which is expressed as follows:

\[
QI_{it} = \frac{N Pub_{it}}{SC_{it}}
\]

\( W \) is a sum of three components as follows:

\[
W = A_i ACT_{it} + B_i y_{it} + C_i QI_{it}
\]

where \( A_i, B_i, \) and \( C_i \) are the given parameters. A higher value of \( A_i \) ensures a higher weight for average early completion, a higher value of \( B_i \) ensures higher weight for the group having own funding for scholarships, while a higher value of \( C_i \) provides higher weight for the group’s quality index. The quality index may include other achievements, such as quality of theses and external recognition.

3. Differential evolution

DE is a powerful global search algorithm for real parameter optimization. It combines the concept of using larger population from a GA and self-adapting mutation from evolution strategy (Storn & Price, 1995). DE differs from other EAs mainly in its generation of new vectors by adding the weighted difference vector between two individuals to a third individual (Storn & Price, 1995). We have selected DE in this paper because of its superior search ability in solving complex practical problems, and it does not require the satisfaction of any mathematical properties of a problem on hand (Sarker & Newton, 2007; Elsayed, Sarker, & Essam, 2013a). Also, the HDRSR problem is dynamic, so using flexible algorithms like DE can guarantee better performance in comparison with deterministic methods (Sarker, Kamruzzaman, & Newton, 2003). The rest of this section gives an overview of DE’s operators and parameters.

3.1. Mutation

A mutant vector is generated by multiplying \( F \) by the difference between two random vectors and the result is added to a third random vector (DE/rand/1) as

\[
\bar{v}_z = \bar{x}_{r_1, z} + F(\bar{x}_{r_2, z} - \bar{x}_{r_3, z})
\]

where \( r_1, r_2, \) and \( r_3 \) are different random integer numbers \( \in [1, PS] \) and none of them is similar to \( z = 1, 2, \ldots, PS \). \( PS \) is the population size. The type of mutation operator has a great effect on the performance of DE. As a consequence, many mutation types have been introduced over the last era,
such as: DE/best/1 (Storn & Price, 1997), DE/rand-to-best/1 (Qin, Huang, & Suganthan, 2009) and DE/current-to-best (Zhang & Sanderson, 2009).

3.2. Crossover

There are two well-known crossover schemes, exponential and binomial. In an exponential crossover, firstly, an integer index, $l$, is randomly selected from a range $[1,n]$, where $n$ is the problem dimension. This index acts as an initial position in the target vector from where an exchange of variables with the donor vector begins. An integer index, $L$, that defines the number of components the donor vector contributes to the target vector, is randomly selected, such that $L \in [1,n]$. Subsequently, a trial vector ($\overline{u}$) is calculated such that

\[
\overline{u}_{z,j} = \begin{cases} 
  v_{z,j} & \text{for } j = (l)_n, (l + 1)_n, \ldots, (l + L - 1)_n \\
  x_{z,j} & \forall j \in [1,n] 
\end{cases}
\]

where $j = 1, 2, \ldots, D$, and $(l)_n$ denotes a modulo function with a modulus of $n$ and a starting location of $l$.

On the other hand, the binomial crossover is conducted on every variable with a predefined crossover probability, such that:

\[
\overline{u}_{z,j} = \begin{cases} 
  v_{z,j} & \text{if } \text{rand} \leq Cr|j = j_{\text{rand}} \\
  x_{z,j} & \text{otherwise}
\end{cases}
\]

where $j_{\text{rand}} \in 1, 2, \ldots, D$ is a randomly selected index, which ensures $\overline{u}_z$ gets at least one component from $\overline{v}_z$.

3.3. Selection

The selection process is simple, in which an offspring will be survived to the next generation, if it is better than its parent, based on its objective value and/or constraints violation.

Over the last two decades, many DE variants have been proposed to adapt DE parameters and/or operators.

Storn and Price (1997) recommended a population size of $5n - 20n$ (and an $F$ value of 0.5, while Rönkkönen (2009) indicated that $F$ is typically between 0.40 and 0.95, with $F = 0.9$ being a good first choice. Abbass (2002) proposed generating $F$ using a Gaussian distribution $N(0, 1)$. This technique was then modified in Elsayed, Sarker, and Essam (2011). Qin et al. (2009) proposed a novel DE algorithm (SaDE), where $F$ was approximated by a normal distribution $N(0.5, 0.3)$, and was truncated to the interval $(0, 2]$. $Cr$ was randomly generated according to an independent normal distribution with mean $Cr_m$ and standard deviation 0.1. The $Cr_m$ values were fixed for five generations before the next regeneration. $Cr_m$ was initialized to 0.5, and it was updated every 25 generations based on the recorded successful $Cr$ values since the last $Cr_m$ update.

Using fuzzy logic controllers, Liu and Lampinen (2005) presented a fuzzy adaptive DE. Brest, Greiner, Boskovic, Mernik, and Zumer (2006) proposed a self-adaptation scheme for the DE control parameters, where in it, a set of $F$ and $Cr$ values were assigned to each individual in the population, thus augmenting the dimensions of each vector. Zhang and Sanderson (2009) introduced an adaptive DE algorithm with optional external memory (JADE). In it, at each generation, $Cr_z$ of each individual was independently generated according to a normal distribution of mean $\mu_{Cr}$ and standard deviation of 0.1. $\mu_{Cr}$ was initialized at a value of 0.5 and was latter updated. Similarly, $F_z$ of each individual was independently generated according to a Cauchy distribution with location parameter ($\mu_F$) and scale parameter 0. $\mu_F$ was initialized at a value of 0.5 and was subsequently updated at the end of each generation.
Sarker, Elsayed, and Ray (2014) proposed a DE algorithm that used a mechanism to dynamically select the best performing combinations of parameters $C_r$ and $F$ for a problem during the course of a single run. The performance of the algorithm was judged by solving three well-known sets of optimization test problems (two constrained and one unconstrained). The results demonstrated that the proposed algorithm was superior to other state-of-the-art algorithms. Elsayed et al. (2011) proposed an algorithm that divides the population into four sub-populations. Each sub-population uses one combination of search operators. During the evolutionary process, the sub-population sizes were adaptively varied, such that the sub-population size of each successful operator was increased, and at the same time the sub-population size of the unsuccessful operators was shrunk. The measure of success and failure of any combination of operators was decided based on changes in the fitness values, constraint violations, and the feasibility ratio of the sub-populations individuals. The algorithm performed well on a set of constrained problems. The algorithm was then extended and improved in Elsayed, Sarker, and Essam (2012, 2013b).

Zamuda and Brest (2012) proposed an algorithm that incorporated two multiple mutation strategies into a self-adaptive DE (jDE) (Brest et al., 2006) and a population reduction methodology which was introduced in Brest and Maučec (2008). The algorithm was tested on 22 real-world applications, and showed better performance than two other algorithms. Brest et al. (2013) also proposed a DE algorithm which embedded a self-adaptation mechanism for parameter control. In it, the population was divided into sub-populations to apply more DE strategies, and a population diversity mechanism was also introduced. The algorithm was tested on a set of unconstrained problems.

### 4. A DE algorithm for HDRSR

The general framework of the DE algorithm used in this research is presented in Algorithm 1.

**Algorithm 1** General framework of DE considered

1. Generate an initial random population. The variables of each individual $(\tilde{x}_j)$ must be within its boundaries;
2. Calculate the fitness value and constraint violation of $(\tilde{x}_j)$;
3. while $\text{iter} < \text{iter}_{\text{max}}$ do
4. for $z = 1 : PS$ do
5. Generate a new individual $(u'_z)$;
6. Calculate the fitness value and constraint violation of $(u'_z)$;
7. if $u'_z$ is better than $\tilde{x}_j$ then
8. $u'_z$ is survived to the next generation;
9. end if
10. Update and sort the new population.
11. end for
12. $\text{iter} \leftarrow \text{iter} + 1$
13. end while

Zamuda and Brest (2012) proposed an algorithm that incorporated two multiple mutation strategies into a self-adaptive DE (jDE) (Brest et al., 2006) and a population reduction methodology which was introduced in Brest and Maučec (2008). The algorithm was tested on 22 real-world applications, and showed better performance than two other algorithms. Brest et al. (2013) also proposed a DE algorithm which embedded a self-adaptation mechanism for parameter control. In it, the population was divided into sub-populations to apply more DE strategies, and a population diversity mechanism was also introduced. The algorithm was tested on a set of unconstrained problems.

Firstly, instead of encoding two initial populations, one for $x$ and one for $y$, each individual of a length $n$, a single population is encoded, of size $PS$, and each individual with a length $2n$, where the first $n$ components represent $x$, while the subsequent $n$ components are for $y$. For simplicity, instead on saying $2n$ as the problem dimension, we name it to $n$. For simplicity, we will use $x$ to represent the decision variables. Each individual must be within its range, such that

$$x_{zj} = x_{zj} + \text{rand} \times (\bar{x}_{zj} - x_{zj})$$

where $x_{zj}$, $\bar{x}_{zj}$ are the lower and upper bounds of the decision variable $x_j$, and rand is a random number $\in [0, 1]$. As we deal with an integer optimization problem, each $x_{zj}$ is rounded to an integer number, as depicted in Figure 1.

Subsequently, DE takes place to generate new individuals. DE/current-to-best (Zhang & Sanderson, 2009) is used, along with the binomial crossover, such that
where $x_{best,j}$ is the $j$th variable of the best individual within the current population, a possible representation of one individual is represented in Figure 2.

If $\vec{u}_{z}$ is better than $\vec{x}_z$, it will survive to the next generation; otherwise keep $\vec{x}_z$ in the next generation. The process continues until a stopping criterion is met. The definition of superiority is based on the superiority of feasible solutions technique (Deb, 2000), as it does not require user-defined parameters. In it, three conditions exist: (1) between two feasible candidates, the fittest one (according to fitness function) is selected; (2) a feasible point is always better than an infeasible one; and (3) between two infeasible solutions, the one with a smaller sum of constraint violations ($\Theta$) is chosen.

5. Experimental results
For the current semester $t$, the optimization model presented in Section 2 is a simple static problem, where all the relevant data and parameters can be calculated and generated using the historical data and based on the goal of the administration. For analyzing multiple periods in the future, some parameters (such as $ACT_{it}$ and $QI_{it}$) are dynamically changed with relation to their earlier activities and performances, and some other parameters (such as $NA_{it}$ and $NS_{t}$) are basically random variables. For multiple periods analysis, we ran the model for a single period, update the parameters, and then re-ran it for the next period. The process continues until all $T$ periods are completed. For the

Table 2. Data generation

| Parameter | Description |
|-----------|-------------|
| $n$       | 5 groups for $x$ and the same is for $y$, hence $n = 10$ |
| $NS_{i,t}$| Randomly generated between 20 and 30 |
| $SS_{i,t}$| Randomly between 0 and 3 |
| $NA_{i,t}$| Randomly between 25 and 40 |
| $MaxSC_{i,t}$| 48 for each group (8 supervisors, with a maximum supervision limit of 6 of each one) |
| $SC_{i,t}$| Randomly between 0 and $MaxSC_{i,t} - SCom_{i,t-1}$ |
| $SCom_{i,t}$| Randomly between 0 and $k_{i,t-6}$ |
| $k_{i,t}$| Randomly between 0 and $MaxSC_{i,t} - SCom_{i,t-1}$ |
| $DES_{kit}$| Randomly generated 5 and 10 |
| $D$ | 3 |
| $QI_{it}$| 1 to 5, where 5 is assigned for the best group. |
| $S$ | 6 |
| $t$ | 7 |
| $T$ | 18 |
| $A$ | 10 |
| $B$ | 5 |
| $x_{z,j}$| $0$ |
| $x_{z,j}$| $NA_{i,t}$ |
In our experimental study, we used random values for some parameters within their ranges observed in the past. Alternatively, the predicted values can be used which can then be changed with the availability of updated information, as shown in Table 2.

Regarding DE parameters, both $F \in [0.4 - 0.95]$, while $Cr$ was set at a value of 0.95 (Sarker et al., 2014). The algorithm was run for 25 times at each period. The best and mean objective values were recorded along with the standard deviation, as shown in Table 3. From this table, it is clear that the algorithm is robust, in which it was able to obtain the same solution, in all 25 runs, at each semester.

Furthermore, Figure 3 shows the number of students that may be recruited during each semester for the subsequent 12 semesters. This figure shows that the numbers of students, in each group, will be close to each other in the long run if they have similar supervision capacity. In the optimization
process, the students will be assigned to high performing groups first and then to the remaining capacity in other groups if the constraints permit.
Figure 5. Aggregated summary of students currently enrolled and recruited in each semester.

Figure 6. Number of students who completed their studies at each semester.

Figure presents two scenarios. In scenario 1, we assume that there are many scholarships in each semester and in scenario 2 only a few scholarships are available in each semester. In Figure 4, we present the number of students recruited in each group, at each semester, and the supervision capacity. From this figure, it is clear that the numbers of students recruited are optimized to be identical.
to the supervision capacity in each group, while Figure shows that the best performing group will receive more students compared to other groups due to the limited number of scholarships.

Figure 5 provides a summary of the total sum of enrolled and recruited students in all groups.

Lastly, Figure 6 is presented to give an overview of the performance of each group, in terms of the number of students who completed their studies at each semester.

5.1. Comparison to other algorithms
In this section, we compare the results obtained with other methods which are well-known in the literature (1) GA, and (2) branch and bound (BB) technique. Both of them are available in Matlab. The results obtained are shown in Table 4, and a comparison summary is presented in Table 5. From the results obtained, it was found that all algorithms were able to obtain the same best results. However, considering the average results achieved, it was shown that DE was the best. This gives a conclusion that DE is more robust than the other two algorithms in solving the problem under consideration in this paper.

Also, the Wilcoxon signed rank test (Corder & Foreman, 2009) is considered to statistically compare between both algorithms. As a null hypothesis, it is assumed that there is no significant difference between the best and/or average results of two samples, while the other hypothesis is that there is a significant difference in the best and/or mean fitness values of the two samples. Using a significance level of 5%, one of three signs (+, −, and ≈) is assigned for the comparison of any two algorithms, where the “+” sign means that the first algorithm is significantly better than the second, the “−” sign means that the first algorithm is significantly worse, and the “≈” sign means that there

| Table 4. Function values obtained at each semester |
|---|---|---|---|---|---|---|
| t  | DE | GA | BB | DE | GA | BB |
| 7  | 2.1362E+02 | 2.1362E+02 | 2.1362E+02 | 2.1362E+02 | 2.1362E+02 | 2.1362E+02 | 2.1362E+02 | 2.1362E+02 | 2.1362E+02 |
| 8  | 1.7674E+02 | 1.7674E+02 | 1.7674E+02 | 1.7674E+02 | 1.7674E+02 | 1.7674E+02 | 1.7674E+02 | 1.7674E+02 | 1.7674E+02 |
| 9  | 2.1210E+02 | 2.1210E+02 | 2.1210E+02 | 2.1210E+02 | 2.1210E+02 | 2.1210E+02 | 2.1210E+02 | 2.1210E+02 | 2.1210E+02 |
| 10 | 1.2655E+02 | 1.2655E+02 | 1.2655E+02 | 1.2655E+02 | 1.2655E+02 | 1.2655E+02 | 1.2655E+02 | 1.2655E+02 | 1.2655E+02 |
| 11 | 1.2205E+02 | 1.2205E+02 | 1.2205E+02 | 1.2205E+02 | 1.2205E+02 | 1.2205E+02 | 1.2205E+02 | 1.2205E+02 | 1.2205E+02 |
| 12 | 1.3513E+02 | 1.3513E+02 | 1.3513E+02 | 1.3513E+02 | 1.3513E+02 | 1.3513E+02 | 1.3513E+02 | 1.3513E+02 | 1.3513E+02 |
| 13 | 1.3941E+02 | 1.3941E+02 | 1.3941E+02 | 1.3941E+02 | 1.3941E+02 | 1.3941E+02 | 1.3941E+02 | 1.3941E+02 | 1.3941E+02 |
| 14 | 1.2122E+02 | 1.2122E+02 | 1.2122E+02 | 1.2122E+02 | 1.2122E+02 | 1.2122E+02 | 1.2122E+02 | 1.2122E+02 | 1.2122E+02 |
| 15 | 1.1663E+02 | 1.1663E+02 | 1.1663E+02 | 1.1663E+02 | 1.1663E+02 | 1.1663E+02 | 1.1663E+02 | 1.1663E+02 | 1.1663E+02 |
| 16 | 1.2524E+02 | 1.2524E+02 | 1.2524E+02 | 1.2524E+02 | 1.2524E+02 | 1.2524E+02 | 1.2524E+02 | 1.2524E+02 | 1.2524E+02 |
| 17 | 1.2930E+02 | 1.2930E+02 | 1.2930E+02 | 1.2930E+02 | 1.2930E+02 | 1.2930E+02 | 1.2930E+02 | 1.2930E+02 | 1.2930E+02 |
| 18 | 1.4790E+02 | 1.4790E+02 | 1.4790E+02 | 1.4790E+02 | 1.4790E+02 | 1.4790E+02 | 1.4790E+02 | 1.4790E+02 | 1.4790E+02 |

| Table 5. A comparison summary of DE against GA and BB |
|---|---|---|---|---|---|
| Algorithms | Results | Better | Similar | Worse | Decision |
| DE vs. GA | Best | 0 | 12 | 0 | ≈ |
| Mean | 12 | 0 | 0 | + |
| DE vs. SQP | Best | 0 | 12 | 0 | ≈ |
| Mean | 7 | 5 | 0 | + |
is no significant difference between the two algorithms. The results are shown in Table 5. From this table, it was found that DE was statistically better than both other algorithms only based on the average results.

6. Conclusions and future work

In this paper, we have introduced an interesting optimization problem arising from HDRSR process and completion in a university environment. The problem is to maximize a user given performance index by satisfying supervision capacity and resource limitation constraints. This is basically a dynamic optimization problem where many parameters not only vary from period to period but also they are difficult to predict in advance. In this research, we have presented a mathematical model to represent this problem and applied DE algorithm to solve the model. Different analysis related to recruitment and completion has been provided. This model can be used for long-run recruitment planning of higher degree research programs based on the forecasted number of applications and scholarships.

For future work, we intend to (1) study the problem in different academic units and institutions, (2) consider different objective functions, (3) consider simultaneous optimization of more than one objective function, and (4) consider many units under an institutions with limited budget constraint.

Funding
The authors received no direct funding for this research.

Author details
Ruhul Sarker
E-mail: r.sarker@adfa.edu.au
Saber Elsayed
E-mail: s.elsayed@adfa.edu.au
1 School of Engineering and Information Technology, University of New South Wales at Canberra, Northcott Drive, Canberra, 2600 Australia.

Citation information
Cite this article as: Evolutionary algorithm for analyzing higher degree research student recruitment and completion, Ruhul Sarker & Saber Elsayed, Cogent Engineering (2015), 2: 1063760.

References
Abbas, H. A. (2002). The self-adaptive pareto differential evolution algorithm. In IEEE Congress on Evolutionary Computation (CEC) (Vol. 1, pp. 831–836). Honolulu, HI: IEEE.
Brest, J., Boskovic, B., Zamuda, A., Fister, I., & Mezura-Montes, E. (2013). Real parameter single objective optimization using self-adaptive differential evolution algorithm with more strategies. In IEEE Congress on Evolutionary Computation (CEC) (pp. 377–383). Cancun: IEEE.
Brest, J., Greiner, S., Boskovic, B., Mernik, M., & Zumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE Transactions on Evolutionary Computation, 10, 646–657.
Brest, J., & Maučec, M. S. (2008). Population size reduction for the differential evolution algorithm. Applied Intelligence, 29, 228–247.
Corder, G. W., & Foreman, D. J. (2009). Nonparametric statistics for non-statisticians: A step-by-step approach. Hoboken, NJ: Wiley.
Deb, K. (2000). An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering, 186, 311–338.
Elsayed, S. M., Sarker, R. A., & Essam, D. L. (2011). Multi-operator based evolutionary algorithms for solving constrained optimization problems. Computers & Operations Research, 38, 1877–1896.
Elsayed, S. M., Sarker, R. A., & Essam, D. L. (2012). On an evolutionary approach for constrained optimization problem solving. Applied Soft Computing, 12, 3208–3227.
Elsayed, S. M., Sarker, R. A., & Essam, D. L. (2013a). Self-adaptive differential evolution incorporating a heuristic mixing of operators. Computational Optimization and Applications, 54, 771–790.
Elsayed, S. M., Sarker, R. A., & Essam, D. L. (2013b). An improved self-adaptive differential evolution algorithm for optimization problems. IEEE Transactions on Industrial Informatics 9, 89–99.
Liu, J., & Lampinen, J. (2005). A fuzzy adaptive differential evolution algorithm. Soft Computing, 9, 448–462.
Qin, A. K., Huang, V. L., & Suganthan, P. N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Transactions on Evolutionary Computation, 13, 398–417.
Rönkkönen, J. (2009). Continuous multimodal global optimization with differential evolution-based methods. Lappeenranta, FL: Lappeenranta University of Technology.
Sarker, R., Elsayed, S., & Roy, T. (2016). Differential evolution with dynamic parameters selection for optimization problems. IEEE Transactions on Evolutionary Computation, 18, 689–707.
Sarker, R., Kamruzzaman, J., & Newton, C. (2003). Evolutionary optimization (evopt): A brief review and analysis. International Journal of Computational Intelligence and Applications, 3, 311–330.
Sarker, R. A., & Newton, C. S. (2007). Optimization modelling: A practical approach. Boca Raton, FL: CRC Press.
Storn, R., & Price, K. (1995). Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. Journal of Global Optimization, 3, 341–359.
Zamuda, A., & Brest, J. (2012). Population reduction differential evolution with multiple mutation strategies in real world industry challenges. In L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. Zadeh, & J. Zurada (Eds.), Swarm and evolutionary computation (pp. 154–161). Zakopane: Springer.
Zhang, J., & Sanderson, A. C. (2009). Jade: Adaptive differential evolution with optional external archive. IEEE Transactions on Evolutionary Computation, 13, 945–958.
