An Immune-Inspired Algorithm for the Set Cover Problem

Ayush Joshi, Jonathan E. Rowe, and Christine Zarges

School of Computer Science, University of Birmingham,
Edgbaston, Birmingham, UK
{axj006,j.e.rowe,c.zarges}@cs.bham.ac.uk

Abstract. This paper introduces a novel parallel immune-inspired algorithm based on recent developments in the understanding of the germinal centre reaction in the immune system. Artificial immune systems are relatively new randomised search heuristics and work on parallelising them is still in its infancy. We compare our algorithm with a parallel implementation of a simple multi-objective evolutionary algorithm on benchmark instances of the set cover problem taken from the OR-library. We show that our algorithm finds feasible solutions faster than the evolutionary algorithm using less parameters and communication effort.

Keywords: Artificial immune systems, GSEMO, set cover.

1 Introduction

Artificial immune systems (AIS) are randomised search heuristics inspired from the immune system of vertebrates [3]. Unlike any other biological system, the immune system has several desirable properties combined together, which make it a great inspiration for the design of randomised search heuristics. Due to properties like diversity, robustness, and memory, algorithms inspired by the immune system have been applied to a large number of applications such as machine learning, security, robotics, and optimisation [3].

As more and more problems in the real world are getting increasingly complex the approaches to solve these are unable to scale and maintain robustness [7,10]. Natural processes on the other hand are robust and perform complicated tasks well, thus it is hoped that understanding and using more detailed ideas from these systems will help us design better systems. In this direction some work has been done by Greensmith [7] on the dendritic cell algorithm but this has been limited to intrusion detection and classification. Sim et al. [15] have proposed a hyper-heuristic called NELLI which learns from changing problem landscapes and has been shown to perform better than single human-designed heuristics.

In recent decades with the advancements in technology parallel architectures and multi-processor systems are becoming more and more common. As a consequence parallelisation of evolutionary algorithms (EA), in order to better utilise available resources and save time, is gaining importance and popularity [11]. EAs are inherently suitable for parallel implementations as operations such as fitness

T. Bartz-Beielstein et al. (Eds.): PPSN XIII 2014, LNCS 8672, pp. 243–251, 2014.
© Springer International Publishing Switzerland 2014

calculations and mutations can be performed on separate processors. Parallel EAs based on island models, where each island is an independent population evolving on a separate processor, have features like inherent diversity. Some communication is required to guide the search process and this is achieved by exchange of solutions between islands.

We propose a novel artificial immune algorithm, *the germinal centre artificial immune system* (GC-AIS) which has been developed based on recent understanding of the germinal centre reaction in the immune system [16]. This new model has interesting properties like a dynamic population of islands and inherent parallelism. We compare it with a parallel version of the global simple evolutionary multi-objective optimiser (GSEMO) [12] on instances of the set cover problem taken from the OR-library [1]. It is shown that the GC-AIS is able to reach the feasible solution region faster than the homogeneous island model with less communication effort as well as less parameters to be set manually.

The outline of the paper is as follows: In Section 2 some preliminary information about the parallel GSEMO is provided along with the problem description. Section 3 gives the description of the GC-AIS model along with its pseudocode. In Section 4 the experimental set-up along with the obtained results are provided. The paper is concluded in Section 5 with a discussion on the observed results and conclusions thereafter.

2 Preliminaries

The set cover problem (SCP) can be defined as follows: Given a *universe set* U consisting of m items and a set S of n subsets of U whose union equals U, the goal is to find the smallest subset of S that covers the whole universe U. More formally:

Definition 1. *Let the set of* m *items* $U := \{u_1, ..., u_m\}$ *denote the universe and let* $S := \{s_1, ..., s_n\}$ *such that* $s_i \subseteq U$ *for* $1 \leq i \leq n$ *and* $\bigcup_{i=1}^{n} s_i = U$*. The unicost set cover problem can be defined as finding a selection* $I \subseteq \{1, 2, ..., n\}$ *such that* $\bigcup_{k \in I} s_k = U$ *with minimum* $|I|$ *.*

SCP is a NP-hard combinatorial optimisation problem with many practical applications, one of the most important being scheduling [2]. A survey of techniques used to solve the set cover problem can be found in [2]. The description of SCP above is a constrained single objective problem where the objective is to find the smallest subset of S which covers the universe and the constraint is that the subset covers the universe.

We convert this to a multi-objective version by using the constraint as a secondary objective [5]. Let vector $A = a_1 a_2 \dots a_n \in \{0, 1\}^n$ denote a solution where $a_i = 1$ if set s_i is in the solution and 0 otherwise. Let N equal the number of sub sets selected in A, and let C equal the number of elements left uncovered in U. The fitness function V for the SCP can now be written as a vector $V = \langle C, N \rangle$.

Multi-objective optimisation [4] is the task of finding optimal solutions to a problem which has several objectives, often competing with each other. A solution is said to dominate another if it is better in at least one objective and has at least the same fitness for the other objectives. In this case it is not always possible to order all individuals in a population since two potential solutions may each be good in a different objective and worse in the others. Therefore, there is not necessarily a unique optimal solution but a set of solutions where each member is not dominated by any other solution in the search space. This set is called the Pareto set and its image in the objective space is called the Pareto front.

The *global simple evolutionary multi-objective optimiser* (GSEMO) [6] is a generalisation of the $(1 + 1)$ EA for multi-objective optimisation. It maintains a set of non-dominated solutions in every iteration. The parallel variant of GSEMO called the homogeneous island model GSEMO based on [12] can be described as a collection of μ islands which are fully connected to each other where each island runs an independent instance of the GSEMO algorithm. We refer to this model as parallel GSEMO (PGSEMO) throughout this paper. This is described in Algorithm 1.

Algorithm 1. Parallel GSEMO based on homogeneous island model [12]

Let P_i^t denote the population in each island i at generation t, μ denote number of islands, and p denote probability of communication.
Initialise $P^0 = \{P_1^0, \ldots, P_\mu^0\}$ where $P_i^0 = \{0^n\}$ for $1 \leq i \leq \mu$. Let $t := 0$.
loop
  for each island i in parallel **do**
    Select an individual x from P_i^t uniformly at random.
    Create offspring x' by mutating x with standard bit mutation, i. e., flip each bit with probability $1/n$.
    if any individual in P_i^t dominates x' **then**
      Leave P_i^t unchanged.
    else
      Remove all individuals dominated by x' from P_i^t and add x' to P_i^t.
    end if
    With probability p send copy of population P_i^t to all $\mu - 1$ neighbours.
    Combine P_i^t with copies of populations received from neighbours.
    Remove all dominated solutions from P_i^t and let $P_i^{t+1} = P_i^t$.
  end for
  Let $t = t + 1$.
end loop

Communication effort can be described as the number of individuals which have been exchanged between the islands. We define the total *communication effort* as the total number of individuals which were transmitted in one run of the algorithm and the *parallel running time* as the number of generations it takes for the algorithm to reach an optimal solution [12].

3 The GC-AIS Algorithm

In the immune system [13] *germinal centres* (GC) are regions where the invading *antigen* (Ag) is presented to the *B cells* (a kind of immune cell) which create *antibodies* (Ab) that in turn bind to the pathogen in order to eradicate it. At the start of the invasion, the number of GCs grows and they try to find the best *Ab* for the pathogen by continuously mutating and selecting the B cells which can bind with the pathogen. Periodically GCs communicate by transmitting their *Ab*s to other GCs. By proliferation, mutation and selection of immune cells this GC reaction is able to produce *Ab*s which can eradicate the pathogen. Towards this stage the number of GCs starts declining.

The exact mechanism of selection in the GC is an active area of research and a new theory forms the basis of our algorithm [16]. According to this work the selection of B cells to be kept alive for proliferation, is maintained by the B cells themselves as they secrete *Ab* which bind with *Ag* and in turn directly compete with other B cells to bind with *Ag*. This is an inter-GC phenomenon as *Ab* from one GC can migrate to others and the competition increases which can lead to disappearance of GCs which can not cope up with the *Ab* from other GCs.

The motivation to apply the GC-AIS to the set cover problem comes from our belief that in an abstract way the immune system tries to solve the set cover problem. Every time the body is invaded by a pathogen, the immune system must produce *Ab*s which are able to bind with the antigen (a partial cover) and must improve this *Ab* by optimisation so that the binding is strong enough to eradicate it (full cover). So if we visualise a possible pathogen binding site as an instance of the universe set, and the binding regions of the B cells as possible solutions, then the immune system tries to solve the problem of finding the best match to the pathogen, by randomised variations in the solutions.

The GC-AIS (see Algorithm 2) starts with one GC which contains one B cell, representing a problem solution. Offspring are created by standard bit mutation of B cells in GCs. In the current version of our algorithm we restrict ourselves to GCs that contain only a single B cell. In every generation there is a migration of *Ab* between GCs, performed by transmitting only the fitness value of the offspring from one GC to another. After migration, dominated solutions are deleted which can lead to the eradication of a GC. The surviving offspring form new GCs. This leads to a model where the number of GCs is dynamic in nature.

The GC-AIS always maintains a set of non-dominated solutions in every generation. A parameter for the number of GCs is not required as the number is dynamic and evolves as the algorithm runs. A preliminary design of the GC-AIS can be found in [9].

4 Experimental Results

In this section we present the results obtained on running the GC-AIS and the PGSEMO on some benchmark test instances of the SCP. The instances are

Algorithm 2. The GC-AIS

Let G^t denote the population of GCs at generation t and g_i^t the i-th GC in G^t.
Create GC pool $G^0 = \{g_1^0\}$ and initialise g_1^0. Let $t := 0$.
loop
  for each GC g_i^t in pool G^t in parallel **do**
    Create offspring y_i of individual g_i^t by standard bit mutation.
  end for
  Add all y_i to G^t, remove all dominated solutions from G^t and let $G^{t+1} = G^t$.
  Let $t = t + 1$.
end loop

selected from the SCP test bed of the OR-library [1] where the instances are grouped into classes based on the size of the problem. One instance each was selected randomly from the 12 problem classes named 4, 5, 6, A, B, C, D, E, RE, RF, RG and RH.

The PGSEMO requires the number of islands and the probability of communication to be set manually while GC-AIS does not require these parameters. As in [12], both algorithms initialise individuals in 0^n. This was done as most problem specific algorithms use this method. For the GC-AIS we observed that starting from a random string gives poor results. The stopping criteria can be based on a fixed budget of generations or letting the algorithms run until a certain desired fitness is achieved.

The probability of communication in the PGSEMO was initially based on the equation $p = \mu/mn$, where μ is the number of islands, p is the probability of communication and m and n describe the problem size. This value gives the best performance guarantee for a complete topology [12]. To estimate the number of islands μ for the PGSEMO, GC-AIS was run for 10,000 generations and it was observed that sufficiently good solutions were achieved. The average of the maximum number of islands in 30 runs was used as the number of islands for PGSEMO. This is done so that a fair comparison can be made in terms of computation resources available to both algorithms. Due to space restrictions it is not possible to include plots for every instance in this paper, key representative plots are provided.

The first set of experiments was performed to analyse the solution quality both algorithms can achieve using a fixed budget of fitness evaluations. A quota of 7500 generations was set as a stopping criteria and average fitness achieved per generation was plotted. This can be seen in Figures 1 and 2. It was observed experimentally that using $p = \mu/mn$ to set p resulted in sub-par performance and experiments were tried with higher rates of communication, $p = 1/n$, $p = 1/m$ and $p = 1/\mu$. These are shown for problem *scp*41 in Figure 1. We observed that having the probability of communication $p = 1/\mu$, so that on average every generation one island communicates gives best results.

Figures 1 and 2 show the fitness achieved per generation in both the GC-AIS and the PGSEMO. The dotted lines depict the average number of sets used and the solid lines depict the average number of uncovered elements per generation.

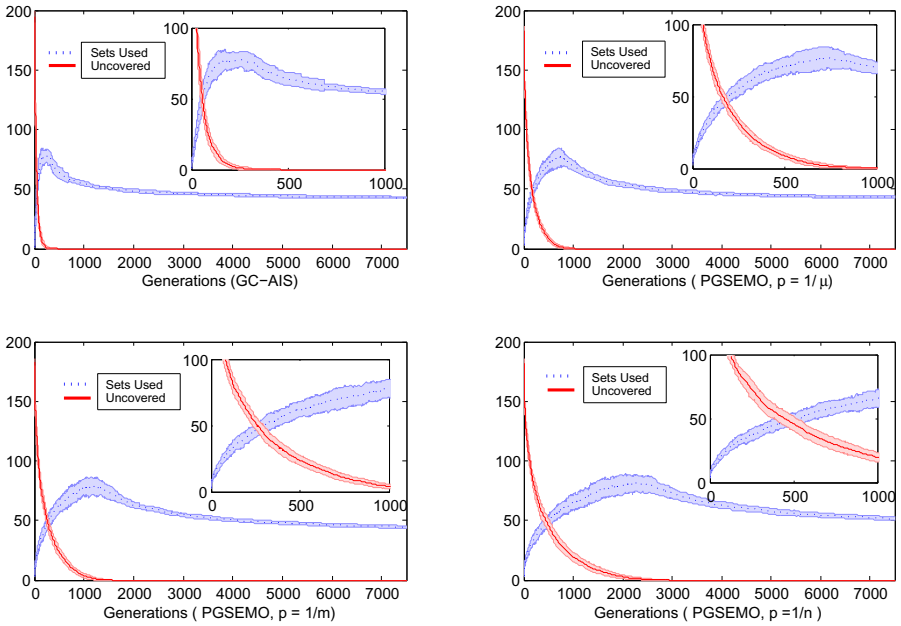

Fig. 1. Fitness plots for GC-AIS and PGSEMO for problem *scp*41 with standard deviation as shaded error bars, averages performed over 30 independent runs

The standard deviation per generation can be seen as the shaded error-bars. To better see the difference between the curves in the early phase, the plots have been zoomed in, which can be seen as the smaller sub-plots inside these figures.

For our next set of experiments we are interested in finding the generations required to reach a fixed fitness value. From our previous experiment we use the best fitness value which has been achieved in every run. Average generations to reach this value were computed and the Wilcoxon rank-sum test was performed. We additionally compare the results achieved by PGESMO and GC-AIS with the best known results and the results of a simple Greedy heuristic [8,14]. All results are shown in Table 1. In 8 out of the 12 rows of the table it can be seen that there is a significant difference (p-value smaller than 0.05) between the performance of the two algorithms, visible from the Wilcoxon rank-sum test results, these entries have been written in bold face. In 7 out of these 8 cases GC-AIS performed faster than PGSEMO.

To estimate the communication effort of the two algorithms, we count the number of individuals which are exchanged between islands per generation. The plots for the accumulated number of communications until generation t are shown in Figure 3 for the problem instances *scp*41 and *scpbnre*4.

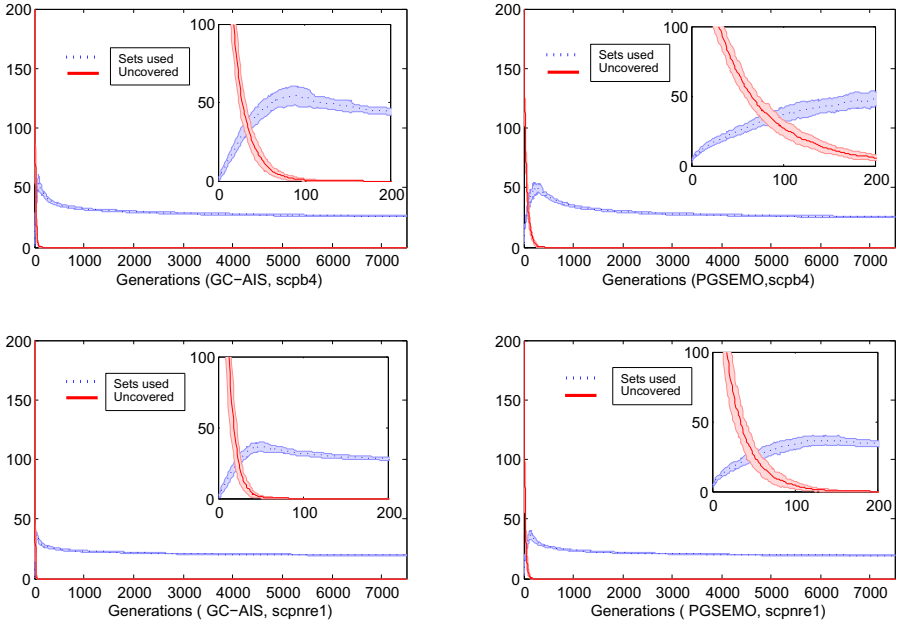

Fig. 2. Fitness plots for GC-AIS and PGSEMO for problem *scpb*4 and *scnpnre*1 with standard deviation as shaded error bars, averages performed over 30 independent runs

Table 1. Generations required to reach a sufficiently good fitness. 'AIS' represents generations required for GC-AIS, 'PGSEMO' represents generations required for PGSEMO, 'fitness' is the target fitness value used as stopping criterion. The column 'WRStest' shows the p-values of the Wilcoxon rank-sum test, 'Gr' contains the fitness achieved by the Greedy heuristic, 'known' contains the best known value and 'achieved' contains best value achieved by GC-AIS in the first set of experiments. Generations are averaged over 30 runs.

Problem	m × n	Fitness	μ	AIS	PGSEMO	WRStest	Gr	Known	Achieved
scp41	200×1000	(0,45)	65	**3654.5**	4349.3	0.0013	41	38	41
scp52	200×2000	(0,43)	65	**3412.4**	4440.4	4.1127e-07	38	34	39
scp63	200×1000	(0,25)	45	2260.8	2037.3	0.3112	21	21	22
scpa5	300×3000	(0,50)	75	**3518.8**	4702.8	9.2603e-09	43	38	44
scpb4	300×3000	(0,28)	50	2941.2	2682.3	0.0575	24	22	25
scpc3	400×4000	(0,58)	90	**3418.**	4765.4	3.6897e-11	47	43	51
scpd2	400×4000	(0,31)	50	3507.8	3450.4	0.9646	26	25	28
scpe1	50×500	(0,5)	12	**961.4**	2051.1	0.0058	5	5	5
scpnre1	500×5000	(0,22)	30	1409	1506	0.1370	18	17	19
scpnrf2	500×5000	(0,12)	20	1867.5	**1490.9**	0.0302	11	10	11
scpnrg3	1000×10000	(0,87)	120	**3168.4**	6519.8	3.0199e-11	-	62	77
scpnrh4	1000×10000	(0,44)	65	**3179**	4069	1.3848e-06	-	34	40

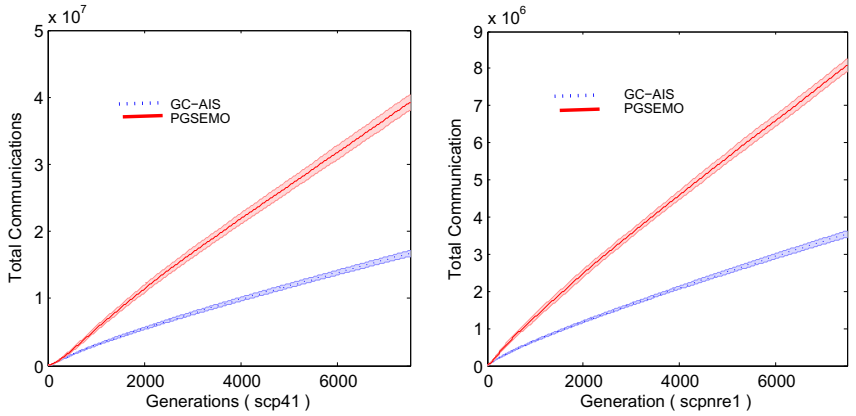

Fig. 3. Total communication cost until generation t, for problems *scp*41 and *scpnre*1 averaged for 30 runs, for GC-AIS and PGSEMO

5 Discussion and Conclusion

The GC-AIS is able to reach the region of feasible solutions, i. e., solutions where the first objective value is 0, faster than the PGSEMO. This can be seen in Figures 1 and 2: in the first 500-1000 generations the solid curve, which depicts the uncovered elements, can be seen to reach 0 faster for GC-AIS than for PGSEMO. It was observed that during the generations towards the end, mutations of individuals in the population very rarely replace any parent. We think that at this stage that all islands in the PGSEMO converged to a similar Pareto set due to communication over the course of the run, while there is in fact just one Pareto set for the GC-AIS. Therefore for the PGSEMO, having many parallel islands each with a similar population increases the chance of finding an improvement, in comparison to a single GC population in the GC-AIS. This advantage of having many islands comes at a cost, which is the communication effort. As communication increases the benefits of parallelism begin to fade, as it becomes a substantial time constraint for the overall performance. The GC-AIS requires far less communications over all than PGSEMO which can be seen in Figure 3. GC-AIS also uses less communication information than the PGSEMO, as only fitness values are sent to other GCs in GC-AIS while the whole population is communicated in PGSEMO.

Parameter setting is a big factor when running an algorithm on a new problem. The GC-AIS has a clear advantage over PGSEMO in terms of parameters to be set: the PGSEMO needs two parameters p and μ to be set manually while GC-AIS does not require any of these. As can be seen in Figure 1, setting the right values for p is crucial to obtain the desired performance. The values we found optimal are different from the ones, which give the best proven performance guarantees, as suggested in [12].

We proposed a novel immune-inspired algorithm called GC-AIS and compared it with a simple multi-objective evolutionary algorithm. With new ideas taken from

the immune system and an interesting motivation to use the set cover problem as a test, we show that the GC-AIS performs faster, uses less communication and has the advantage of not requiring as much human intervention to set it up. In the future we will investigate how the GC-AIS performs on other problem classes and compare it with state-of-the-art techniques for these problems.

References

1. Beasley, J.E.: OR-library: Distributing test problems by electronic mail. The Journal of the Operational Research Society 41(11), 1069–1072 (1990), https://files.nyu.edu/jeb21/public/jeb/info.html
2. Caprara, A., Toth, P., Fischetti, M.: Algorithms for the set covering problem. Annals of Operations Research 98(1-4), 353–371 (2000)
3. De Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer (2002)
4. Deb, K.: Multi-objective Optimization Using Evolutionary Algorithms. Wiley-Blackwell (2001)
5. Friedrich, T., He, J., Hebbinghaus, N., Neumann, F., Witt, C.: Approximating covering problems by randomized search heuristics using multi-objective models. Evolutionary Computation 18(4), 617–633 (2010)
6. Giel, O., Wegener, I.: Evolutionary algorithms and the maximum matching problem. In: Alt, H., Habib, M. (eds.) STACS 2003. LNCS, vol. 2607, pp. 415–426. Springer, Heidelberg (2003)
7. Greensmith, J.: The Dendritic Cell Algorithm. PhD thesis, University of Nottingham (2007), http://www.cs.nott.ac.uk/~jqg/thesis.pdf
8. Grossman, T., Wool, A.: Computational experience with approximation algorithms for the set covering problem. European Journal of Operational Research 101(1), 81–92 (1997)
9. Joshi, A.: Design of a parallel immune algorithm based on the germinal center reaction. In: Proc of GECCO Companion, pp. 1671–1674. ACM (2013)
10. Kim, J., Bentley, P.J.: Towards an artificial immune system for network intrusion detection: An investigation of clonal selection with a negative selection operator. In: Proc. of CEC, vol. 2, pp. 1244–1252. IEEE Press (2002)
11. Luque, G., Alba, E.: Parallel Genetic Algorithms: Theory and Real World Applications. Springer (2011)
12. Mambrini, A., Sudholt, D., Yao, X.: Homogeneous and heterogeneous island models for the set cover problem. In: Coello, C.A.C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) PPSN 2012, Part I. LNCS, vol. 7491, pp. 11–20. Springer, Heidelberg (2012)
13. Murphy, K.: Janeway's Immunobiology. Garland Science (2011)
14. Musliu, N.: Local search algorithm for unicost set covering problem. In: Ali, M., Dapoigny, R. (eds.) IEA/AIE 2006. LNCS (LNAI), vol. 4031, pp. 302–311. Springer, Heidelberg (2006)
15. Sim, K., Hart, E., Paechter, B.: A lifelong learning hyper-heuristic method for bin packing. Evolutionary Computation (to appear, 2014), http://dx.doi.org/10.1162/EVCO_a_00121
16. Zhang, Y., Meyer-Hermann, M., George, L.A., Figge, M.T., Khan, M., Goodall, M., Young, S.P., Reynolds, A., Falciani, F., Waisman, A., Notley, C.A., Ehrenstein, M.R., Kosco-Vilbois, M., Toellner, K.-M.: Germinal center B cells govern their own fate via antibody feedback. The Journal of Experimental Medicine 210(3), 457–464 (2013)