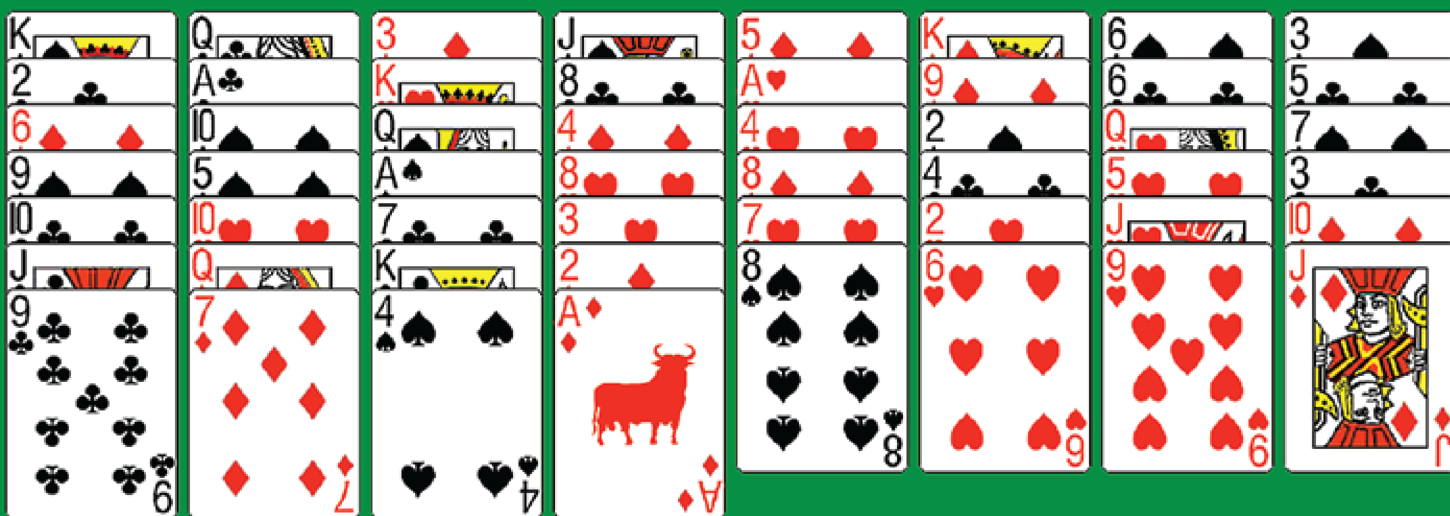
SIGEVOlution

newsletter of the ACM Special Interest Group on Genetic and Evolutionary Computation

Volume 6
Issue 3-4



in this issue

Editorial

I packed my bag and I am now ready to go — Vancouver here I come! I did not attend GECCO since 2011 and I missed it a lot so I really look forward to this trip. Time to meet some very good friends, check the most recent development in the field, and immerse myself into the vibe that makes every GECCO always special.

This issue ends volume six and we are working to catch up with the delay, hopefully in the next 6-12 months. But to succeed in our endeavor, we need your help! We need interesting articles that can show all the amazing stuff that evolutionary computation can do, the awesome applications we developed, the results we achieved, our best theses, and so on.

And to start the showcase there is nothing better than Moshe Sipper, a researchers with an incredible list of Humies medals. In his article with Achiya Elyasaf, Moshe shows us how he used evolutionary computation to build human-competitive FreeCell solvers. If you like the topic, you should definitively download his ebook "Evolved to Win" where the topic is discussed at length, or (even better) pay a few bucks and get the nice printed edition from Lulu.

The awesomeness continues with William Langdon's article about the new features of the GP bibliography, based on the most amazing BibTeXfile you can find in the field, a real goldmine with 9585 GP papers that now includes graphical displays of recent Internet-based paper download activity, of centers of GP expertise, an updated list of new papers, and a blog. Bill did an amazing job along the years and the bibliography is an incredible support to the community.

If you could not go to EvoStar this year you might get an update from Justyna Petke's report and, just in case, check out the amazing flyer for the 2015 Evostar call for papers.

As always, I owe my thanks to the many people who helped me in this: Moshe Sipper, Achiya Elyasaf, William B. Langdon, Justyna Petke, Daniele Loiacono, Cristiana Bolchini, Viola Schiaffonati, Francesco Amigoni, and Franz Rothlauf.

If you wonder where the next GECCO will be, the cover provides some hints.

See you in Vancouver!

Pier Luca
July 10, 2014

SIGEVOlution Volume 6, Issue 3-4

Newsletter of the ACM Special Interest Group on Genetic and Evolutionary Computation.

Contributors to this Issue

Moshe Sipper
Achiya Elyasaf
William B. Langdon
Justyna Petke

Contents

ISSN: 1931-8499

Lunch Isn't Free — But Cells Are

Evolving FreeCell Players

Moshe Sipper (sipper@cs.bgu.ac.il) and Achiya Elyasaf (achiya.e@gmail.com)
Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel.

The application of computational intelligence techniques within the vast domain of games has been increasing at a breathtaking speed. Over the past several years our research group has produced a plethora of results in numerous games of different natures, evidencing the success and efficiency of evolutionary algorithms in general — and genetic programming in particular — at producing top-notch, human-competitive game strategies. Herein, we describe our study of the game of FreeCell, which produced two Gold Humie Awards. Our top evolved FreeCell player is the best published player to date, able to convincingly beat high-ranking human players.

1 Let's Play!

Ever since the dawn of artificial intelligence (AI) in the 1950s games have been part and parcel of this lively field. In 1957, a year after the Dartmouth Conference that marked the official birth of AI, Alex Bernstein designed a program for the IBM 704 that played two amateur games of chess. In 1958, Allen Newell, J. C. Shaw, and Herbert Simon introduced a more sophisticated chess program (beaten in thirty-five moves by a ten-year-old beginner in its last official game played in 1960). Arthur L. Samuel of IBM spent much of the fifties working on game-playing AI programs, and by 1961 he had a checkers program that could play rather decently. In 1961 and 1963 Donald Michie described a simple trial-and-error learning system for learning how to play tic-tac-toe (or Noughts and Crosses) called MENACE (for Matchbox Educable Noughts and Crosses Engine) [24].

Why do games attract such interest? "There are two principal reasons to continue to do research on games," wrote Epstein [6]. "First, human fascination with game playing is long-standing and pervasive. Anthropologists have catalogued popular games in almost every culture . . . Games intrigue us because they address important cognitive functions . . . The second reason to continue game-playing research is that some difficult games remain to be won, games that people play very well but computers do not. These games clarify what our current approach lacks. They set challenges for us to meet, and they promise ample rewards."

Studying games may thus advance our knowledge both in cognition and artificial intelligence, and, last but not least, games possess a competitive angle that coincides with our human nature, thus motivating both researcher and student alike.

During the past few years there has been an ever-increasing interest in the application of computational intelligence techniques in general, and evolutionary algorithms in particular, within the vast domain of games. The year 2005 saw the first *IEEE Symposium on Computational Intelligence and Games*, which went on to become an annually organized event. The symposia's success and popularity led to their promotion from symposium to conference in 2010, and also spawned the successful journal *IEEE Transactions on Computational Intelligence and AI in Games* in 2009.

Clearly, there's a serious side to games [24]. In this article we shall recount our success in tackling the popular game of FreeCell, an endeavor that garnered two Gold Humie Awards (in 2011 and 2013).

2 FreeCell

Discrete puzzles, also known as single-player games, are an excellent problem domain for artificial intelligence research, because they can be parsimoniously described yet are often hard to solve [20]. As such, puzzles have been the focus of substantial research in AI during the past decades (e.g., Hearn [11], Robertson and Munro [22]). Nonetheless, quite a few NP-Complete puzzles have remained relatively neglected by academic researchers (see [16] for a review).

Search algorithms for puzzles (as well as for other types of problems) are strongly based on the notion of approximating the distance of a given configuration (or *state*) to the problem's solution (or *goal*). Such approximations are found by means of a computationally efficient function, known as a *heuristic function*. By applying such a function to states reachable from the current one considered, it becomes possible to select more-promising alternatives earlier in the search process, possibly reducing the amount of search effort (typically measured in number of nodes expanded) required to solve a given problem. The putative reduction is strongly tied to the quality of the heuristic function used: employing a perfect function means simply "strolling" onto the solution (i.e., no search de facto), while using a bad function could render the search less efficient than totally uninformed search, such as breadth-first search (BFS) or depth-first search (DFS).

A well-known, highly popular example within the domain of discrete puzzles is the card game of FreeCell. Starting with all cards randomly divided into k piles (called *cascades*), the objective of the game is to move all cards onto four different piles (called *foundations*) — one per suit — arranged upwards from the ace to the king. Additionally, there are initially empty cells (called *free cells*), whose purpose is to aid with moving the cards. Only exposed cards can be moved, either from free cells or cascades. Legal move destinations include: a home (foundation) cell, if all previous (i.e., lower) cards are already there; empty free cells; and, on top of a next-highest card of opposite color in a cascade (Figure 1). FreeCell was proven by Helmert [13] to be NP-complete. In his paper, Helmert explains that the hardness of the domain is not (or at least not exclusively) due to the difficulty in allocating free cells or empty pile positions, but rather due to the choice of which card to move on top of a pile when there are two possible choices.

Computational complexity aside, even in its limited popular version (described below) many (oft-frustrated) human players (including the authors) will readily attest to the game's hardness. The attainment of a competent machine player would undoubtedly be considered a human-competitive result.

FreeCell remained relatively obscure until it was included in the Windows 95 operating system (and in all subsequent versions), along with 32,000 problems — known as *Microsoft 32K* — all solvable but one (this latter, game #11982, was proven to be unsolvable [12]). Due to Microsoft's move, FreeCell has been claimed to be one of the world's most popular games [1]. The Microsoft version of the game comprises a standard deck of 52 cards, 8 cascades, 4 foundations, and 4 free cells. Though limited in size, this FreeCell version still requires an enormous amount of search, due both to long solutions and to large branching factors. Thus it remains out of reach for optimal heuristic search algorithms, such as A* and iterative deepening A* [8; 17], both considered standard methods for solving difficult single-player games (e.g., Junghanns and Schaeffer [15], Korf [19]). FreeCell remains intractable even when powerful enhancement techniques are employed, such as transposition tables [7; 25] and macro moves [18].

Despite there being numerous FreeCell solvers available via the Web, few have been written up in the scientific literature. The best published solvers to date are our own GA-based solver, winner of a Gold Humie in 2011 [3; 5; 24], and our GP-based solver, winner of a Gold Humie in 2013 [4]. Using a standard GA, we were able to outperform the previous top gun — Heineman's staged deepening algorithm (HSD) — which is based on a hybrid A* / hill-climbing search algorithm. Later, using GP, we beat our own GA-based FreeCell player. We shall focus herein on this latter top gun.

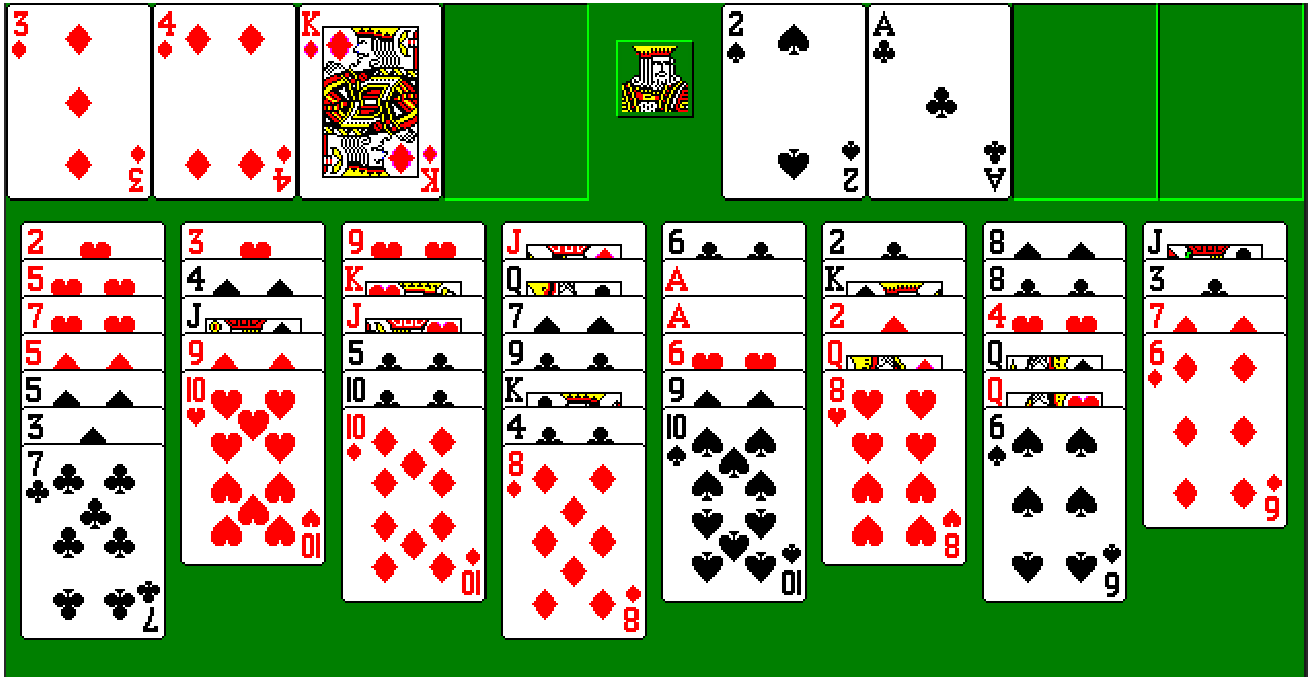


Fig. 1: A FreeCell game configuration. Cascades: Bottom 8 piles. Foundations: 4 upper-right piles. Free cells: 4 upper-left cells. Note that cascades are not arranged according to suits, but foundations are. Legal moves for the current configuration: 1) moving 7♣ from the leftmost cascade to either the pile fourth from the left (on top of the 8♢), or to the pile third from the right (on top of the 8♡); 2) moving the 6♢ from the right cascade to the left one (on top of the 7♣); and 3) moving any single card on top of a cascade onto the empty free cell.

3 Search Algorithms

3.1 Iterative Deepening

We initially implemented standard iterative deepening search [17] as the heart of our game engine. This algorithm may be viewed as a combination of DFS and BFS: starting from a given configuration (e.g., the initial state), with a minimal depth bound, we perform a DFS search for the goal state through the graph of game states (in which vertices represent game configurations, and edges — legal moves). Thus, the algorithm requires only $\theta(n)$ memory, where n is the depth of the search tree. If we succeed, the path is returned. If not, we increase the depth bound by a fixed amount, and restart the search.

Note that since the search is incremental, when we find a solution we are guaranteed that it is optimal since a shorter solution would have been found in a previous iteration (more precisely, the solution is optimal or near optimal, depending on whether the depth increase equals 1 or is greater than 1). For difficult problems such as FreeCell finding *a* solution is sufficient, and there is typically no requirement of finding the optimal solution.

An iterative deepening-based game engine receives as input a FreeCell initial configuration (known as a deal), as well as some run parameters, and outputs a solution (i.e., a list of moves) or an indication that the deal could not be solved.

We observed that the search algorithm did not find a solution in a timely fashion even when allowed to use all the available memory in order to eliminate revisiting nodes (2GB in our case, as opposed to [10] where the node count was limited). Virtually all Microsoft 32K problems could not be solved, hence we concluded that heuristics were essential for solving FreeCell instances because uninformed search alone was insufficient.

3.2 Iterative Deepening A*

Given that the HSD solver outperforms all other solvers (except ours), we implemented the heuristic function used by HSD along with the iterative deepening A* (IDA*) search algorithm [17], one of the most prominent methods for solving puzzles (e.g., Junghanns and Schaeffer [15], Korf [19], Samadi et al. [23]). This algorithm operates similarly to iterative deepening, except that in the DFS phase heuristic values are used to determine the order by which children of a given node are visited. This move ordering is the only phase wherein the heuristic function is used — the open list structure is still sorted according to depth alone.

IDA* underperformed where FreeCell was concerned, unable to solve many instances (deals). Even using several heuristic functions, IDA* — despite its success in other difficult domains — yielded inadequate performance: less than 1% of the deals we tackled were solved in a reasonable time.

At this point we opted for employing the HSD solver in its entirety, rather than merely the HSD heuristic function.

3.3 Staged Deepening

Heineman's Staged Deepening (HSD) algorithm is based on the observation that there is no need to store the entire search space seen so far in memory. This is so because of a number of significant characteristics of FreeCell:

- For most states there is more than one distinct permutation of moves creating valid solutions. Hence, very little backtracking is needed.
- There is a relatively high percentage of irreversible moves: according to the game's rules a card placed in a home cell cannot be moved again, and a card moved from an unsorted pile cannot be returned to it.
- If we start from game state s and reach state t after performing k moves, and k is large enough, then there is no longer any need to store the intermediate states between s and t. The reason is that there is a solution from t (first characteristic) and a high percentage of the moves along the path are irreversible anyway (second characteristic).

Thus, the HSD algorithm may be viewed as two-layered IDA* with periodic memory cleanup [4]. The two layers operate in an interleaved fashion: 1) At each iteration, a local DFS is performed from the head of the open list up to depth k, with no heuristic evaluations, using a transposition table — storing visited nodes — to avoid loops; 2) Only nodes at *precisely* depth k are stored in the open list, which is sorted according to the nodes' heuristic values. In addition to these two interleaved layers, whenever the transposition table reaches a predetermined size, it is emptied entirely, and only the open list remains in memory (see Elyasaf et al. [4] for further details).

When we ran the HSD solver it solved 96% of Microsoft 32K, as reported by Heineman.

3.4 Heuristics and Advisors

At this point we were at the limit of the current state-of-the-art for FreeCell, and we turned to evolution to attain better results. However, we first needed to develop additional heuristics for this domain — which we did. For example, the heuristic `NumWellPlaced` counts the number of well-placed cards in cascade piles, where a pile of cards is well placed if all its cards are in descending order and alternating colors; the heuristic `NumCardsNotAtFoundations` counts the number of cards that are not at the foundation piles. The full list of heuristics can be found in [4].

Apart from heuristics, which estimate the distance to the goal, we also defined *advisors* (or auxiliary functions), incorporating domain features, i.e., functions that do not provide an estimate of the distance to the goal but which are nonetheless beneficial in a GP setting. For example, `IsMoveToCascade` is a Boolean function that examines the destination of the last move and returns true if it was a cascade. Again, the full list of advisors is given in [4].

Experiments with the heuristics demonstrated that each one separately (except for HSDH) was not good enough to guide search for this difficult problem. Thus we turned to evolution.

4 Evolving Heuristics for FreeCell

Combining several heuristics to get a more accurate one is considered one of the most difficult problems in contemporary heuristics research [23; 2].

This task typically involves solving three major sub-problems:

1. How to combine heuristics by *arithmetic* means, e.g., by summing their values or taking the maximal value.
2. Finding exact conditions (i.e., *logic* functions) regarding *when* to apply each heuristic, or combinations thereof — some heuristics may be more suitable than others when dealing with specific game configurations.
3. Finding the proper set of game configurations in order to facilitate the learning process while avoiding pitfalls such as overfitting.

The problem of combining heuristics is difficult mainly because it entails traversing an extremely large search space of possible numeric combinations, logic conditions, and game configurations. To tackle this problem we turned to evolution.

In order to properly solve these three sub-problems, we designed a large set of experiments using three different evolutionary methods, all involving hyper-heuristics: a standard GA, standard (Koza-style) GP, and policy-based GP. Each type of hyper-heuristic was paired with three different learning settings: Rosin-style coevolution, Hillis-style coevolution, and a novel method which we called gradual difficulty. While details of all our experiments can be found in [4] we briefly describe the notion of *policy*, on which our emergent winner is based.

We first introduced policies in [10], where we studied the game of Rush Hour. A policy has the form:

$RULE_1$: IF $Condition_1$ THEN $Value_1$
.
.
$RULE_N$: IF $Condition_N$ THEN $Value_N$
$DEFAULT$: $Value_{N+1}$,

where $Condition_i$ and $Value_i$ represent conditions and estimates, respectively.

Policies are used by the search algorithm in the following manner: The rules are ordered such that we apply the first rule that "fires" (meaning its condition is true for the current state being evaluated), returning its $Value$ part. If no rule fires, the value is taken from the last (default) rule: $Value_{N+1}$. Thus individuals, while in the form of policies, are still heuristics — the value returned by the activated rule is an arithmetic combination of heuristic values, and is thus a heuristic value itself. This accords with our requirements: rule ordering and conditions control when we apply a heuristic combination, and values provide the combinations themselves.

Thus, with N being the number of rules used, each individual in the evolving population contains N $Condition$ GP trees and $N+1$ $Value$ sets of weights used for computing linear combinations of heuristic values. After experimenting with several sizes of policies we settled on $N = 5$, providing us with enough rules per individual, while avoiding cumbersome individuals with too many rules. The depth limit used for the $Condition$ trees was empirically set to 5.

For $Condition$ GP trees, the function set included the functions $\{AND, OR, \leq, \geq\}$, and the terminal set included all the heuristics and auxiliary functions we defined. The sets of weights appearing in $Value$s all lie within the range $[0,1]$, and correspond to the heuristics. All the heuristic values are normalized to within the range $[0,1]$.

Again, while [4] provides the full details of the evolutionary setup, we wish to recount here a major component — fitness computation. An evolving individual's (i.e., FreeCell solver's) fitness score was obtained by running the HSD solver on deals taken from a training set, with the individual used as the heuristic function. Fitness equaled the average search-node reduction ratio. This ratio was obtained by comparing the reduction in number of search nodes — averaged over solved deals — with the average number of nodes when searching with the original HSD heuristic. We experimented with several types of evolution and coevolution, finally finding that Hillis-style coevolution worked best, involving two coevolving populations: solvers and sets of deals to be solved.

5 Major Results

A plethora of results and analyses can be found in [4]. Herein we summarize what we believe to be the major results. Compared to HSDH, GA-FreeCell [3] and Policy-FreeCell reduced the amount of search by more than 78%, solution time by more than 93%, and solution length by more than 30% (with unsolved problems excluded from the count). In addition, Policy-FreeCell solved 99.65% of Microsoft 32K, thus outperforming both HSDH and GA-FreeCell.

How does our evolution-produced player fare against humans? A major FreeCell website[1] provides a ranking of human FreeCell players, listing solution times and win rates (alas, no data on number of deals examined by humans, nor on solution lengths). Since statistics regarding players who played sparsely are not reliable, we focused on humans who played over 30K games — a figure commensurate with our own.

The site statistics, which we downloaded on December 13, 2011, included results for 83 humans who met the minimal-game requirement — all but two of whom exhibited a win rate greater than 91%. Sorted according to the number of games played, the no. 1 player played 160,237 games, achieving a win rate of 96.02%. This human is therefore pushed to the fourth position, with our top player (99.65% win rate) taking the first place, our GA-FreeCell taking the second place, and HSDH coming in third.

When sorted according to average solving time, the fastest human player with win rate above 90% solved deals in an average time of 104 seconds and achieved a win rate of 96.56%. This human is therefore pushed to the fourth position, with HSDH in the third place, GA-FreeCell in the second place, and Policy-FreeCell taking the first place. Note that the fastest human player takes 67 seconds on average to reach a solution. HSDH reduces this average time by 34.3%, while our evolved solvers reduce the average time by 95.5%. These values suggest that outperforming human players in time-to-solve is not a trivial task for a computer. Yet, our evolved solvers manage to shine with respect to time as well.

[1] http://www.freecell.net

Tab. 1: The top three human players (when sorted according to win rate), compared with HSDH, GA-FreeCell, and Policy-FreeCell. Shown are number of deals played, average time (in seconds) to solve, and percent of solved deals from Microsoft 32K. Table arranged in descending order of win rate (percentage of solved deals).

Rank	Name	Deals played	Time	Solved
1	Policy-FreeCell	32,000	3	99.65%
2	JonnieBoy	39,102	270	99.33%
3	time.waster	37,286	191	99.20%
4	Nat_King_C.	54,599	207	98.97%
...				
11	GA-FreeCell	32,000	3	98.36%
...				
66	HSDH	32,000	44	96.43%

If the statistics are sorted according to win rate then our Policy-FreeCell player takes the first place with a win rate of 99.65%, while GA-FreeCell attains the respectable 11th place. Either way, it is clear that when compared with strong, persistent, and consistent humans, Policy-FreeCell emerges as the new best player to date, leaving HSDH far behind (Table 1).

6 Concluding Remarks

Although policies can be seen as a special case of GP trees they yielded good results for this domain while GP did not. A possible reason for this is that the policy structure is more apt for this type of problems. The policy conditions classify states while the values combine the available heuristics. When standard tree-GP is used, the structure is not clear and many meaningless trees are generated.

The heuristics and advisors used as building blocks for the evolutionary process are intuitive and straightforward to implement and compute. Yet, our evolved solvers are the top solvers for the game of FreeCell, suggesting that in some domains good solvers can be achieved with minimal domain knowledge and without the use of much domain expertise. It should be noted that complex heuristics and memory-consuming heuristics (e.g., landmarks and pattern databases) can be easily used as building blocks as well. Such solvers might outperform the simpler ones at the expense of increased run time or code complexity.

There are a number of possible extensions to our work, including:

1. It is possible to implement FreeCell macro moves and thus decrease the search space. Implementing macro moves will yield better results, and we believe that we might even solve the entire Microsoft 32K (excluding unsolvable game #11982).
2. Using complex heuristics, as noted above.
3. The HSD algorithm, enhanced with evolved heuristics, is more efficient than the original version. This is evidenced both by the amount of search reduction and the increased number of solved deals. It remains to be determined whether the algorithm, when aided by evolution, can outperform other widely used algorithms (such as IDA*) in different domains. The fact that the algorithm is based on several properties of search problems, such as the high percentage of irreversible moves and the small number of deadlocks, already points the way towards several domains. A good candidate may be the Satellite game, previously studied in [9; 14].
4. Handcrafted heuristics may themselves be improved by evolution. This could be done by breaking them into their elemental components and evolving their combinations thereof.
5. Many single-agent search problems fall within the framework of AI-planning problems (e.g., with ADL [21]). However, using evolution in conjunction with these techniques is not trivial and may require the use of techniques such as GP policies [10].

It would seem that after attacking FreeCell with our evolutionary guns, human players might wish to heed the words of Yoda: “If no mistake have you made, yet losing you are . . . a different game you should play.”

References

[1] F. Bacchus. AIPS’00 planning competition. *AI Magazine*, 22(1):47–56, 2001.

[2] Edmund K. Burke, Matthew Hyde, Graham Kendall, Gabriela Ochoa, Ender Ozcan, and John R. Woodward. A classification of hyper-heuristic approaches. In M. Gendreau and J-Y. Potvin, editors, *Handbook of Meta-Heuristics 2nd Edition*, pages 449–468. Springer, 2010.

[3] Achiya Elyasaf, Ami Hauptman, and Moshe Sipper. GA-FreeCell: Evolving Solvers for the Game of FreeCell. In N. Krasnogor et al., editors, *GECCO ’11: Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 1931–1938, Dublin, Ireland, 12-16 July 2011. ACM.

[4] Achiya Elyasaf, Ami Hauptman, and Moshe Sipper. Evolutionary design of FreeCell solvers. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(4):270–281, 2012.

[5] Achiya Elyasaf, Yael Zaritsky, Ami Hauptman, and Moshe Sipper. Evolving solvers for FreeCell and the sliding-tile puzzle. In Daniel Borrajo, Maxim Likhachev, and Carlos Linares López, editors, *Proceedings of the Fourth Annual Symposium on Combinatorial Search, SOCS 2011*. AAAI Press, 2011.

[6] S. L. Epstein. Game playing: The next moves. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pages 987–993. AAAI Press, Menlo Park, California USA, 1999.

[7] P. W. Frey. *Chess Skill in Man and Machine*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1979.

[8] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for heuristic determination of minimum path cost. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, February 1968.

[9] Patrik Haslum, Blai Bonet, and Hector Geffner. New admissible heuristics for domain-independent planning. In Manuela M. Veloso and Subbarao Kambhampati, editors, *AAAI ’05: Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, pages 1163–1168. AAAI Press / The MIT Press, 2005.

[10] Ami Hauptman, Achiya Elyasaf, Moshe Sipper, and Assaf Karmon. GP-Rush: using genetic programming to evolve solvers for the Rush Hour puzzle. In *GECCO'09: Proceedings of 11th Annual Conference on Genetic and Evolutionary Computation Conference*, pages 955–962, New York, NY, USA, 2009. ACM.

[11] R. A. Hearn. *Games, Puzzles, and Computation*. PhD thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, 2006.

[12] G. T. Heineman. Algorithm to solve FreeCell solitaire games, January 2009. http://broadcast.oreilly.com/2009/01/january-column-graph-algorithm.html. Blog column associated with the book "Algorithms in a Nutshell book," by G. T. Heineman, G. Pollice, and S. Selkow, O'Reilly Media, 2008.

[13] M. Helmert. Complexity results for standard benchmark domains in planning. *Artificial Intelligence*, 143(2):219–262, 2003.

[14] Malte Helmert. *Understanding Planning Tasks: Domain Complexity and Heuristic Decomposition*, volume 4929 of *Lecture Notes in Computer Science*. Springer, 2008.

[15] A. Junghanns and J. Schaeffer. Sokoban: A challenging single-agent search problem. In *Workshop on Using Games as an Experimental Testbed for AI Research, Proceedings IJCAI-97*, pages 27–36, 1997.

[16] G. Kendall, A. Parkes, and K. Spoerer. A survey of NP-complete puzzles. *ICGA Journal*, 31:13–34, 2008.

[17] R. E. Korf. Depth-first iterative-deepening: An optimal admissible tree search. *Artificial Intelligence*, 27(1):97–109, 1985.

[18] R. E. Korf. Macro-operators: A weak method for learning. *Artificial Intelligence*, 26:35–77, 1985.

[19] Richard E. Korf. Finding optimal solutions to Rubik's cube using pattern databases. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence*, AAAI'97/IAAI'97, pages 700–705. AAAI Press, 1997.

[20] J. Pearl. *Heuristics*. Addison–Wesley, Reading, Massachusetts, 1984.

[21] E. Pednault. ADL: Exploring the middle ground between STRIPS and the situation calculus. In *Proceedings of 1st international conference on Principles of Knowledge Representation and Reasoning*, pages 324–332, 1989.

[22] E. Robertson and I. Munro. NP-completeness, puzzles and games. *Utilas Mathematica*, 13:99–116, 1978.

[23] M. Samadi, A. Felner, and J. Schaeffer. Learning from multiple heuristics. In Dieter Fox and Carla P. Gomes, editors, *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI 2008)*, pages 357–362. AAAI Press, 2008.

[24] Moshe Sipper. *Evolved to Win*. Lulu, 2011. available at http://www.lulu.com/.

[25] Larry A. Taylor and Richard E. Korf. Pruning duplicate nodes in depth-first search. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, AAAI'93, pages 756–761. AAAI Press, 1993.

About the authors

Moshe Sipper is a Professor of Computer Science at Ben-Gurion University of the Negev, Israel. He received his B.A. degree from the Technion — Israel Institute of Technology, and his M.Sc. and Ph.D. degrees from Tel Aviv University, all in Computer Science. During the years 1995–2001 he was a Senior Researcher at the Swiss Federal Institute of Technology in Lausanne. Dr. Sipper's current research focuses on evolutionary computation, mainly as applied to software development and games. At some point or other he also did research in the following areas: bio-inspired computing, cellular automata, cellular computing, artificial self-replication, evolvable hardware, artificial life, artificial neural networks, fuzzy logic, and robotics. Dr. Sipper has published over 160 scientific publications including three research-related books: Evolved to Win, Machine Nature: The Coming Age of Bio-Inspired Computing, and Evolution of Parallel Cellular Machines: The Cellular Programming Approach. He is an Associate Editor of the IEEE Transactions on Computational Intelligence and AI in Games and Genetic Programming and Evolvable Machines, an Editorial Board Member of Memetic Computing, and a past Associate Editor of the IEEE Transactions on Evolutionary Computation. Dr. Sipper won the 1999 EPFL Latsis Prize, the 2008 BGU Toronto Prize for Academic Excellence in Research, and six HUMIE Awards — Human-Competitive Results Produced by Genetic and Evolutionary Computation (Gold, 2013; Gold, 2011; Bronze, 2009; Bronze, 2008; Silver, 2007; Bronze, 2005).

Moshe Sipper is also the author of three novels: Daniel Max and the King in the Tower, Xor: The Shape of Darkness, and The Peaceful Affair. He writes short-short science fiction and fantasy stories, available at To Make a Long Story Short.

Homepage: http://www.moshesipper.com/
Email: sipper@cs.bgu.ac.il

Achiya Elyasaf received the B.Sc. degree (summa cum laude) and the M.Sc. degree (cum laude), both in computer science, from Ben-Gurion University of the Negev, Israel, where he is currently pursuing the Ph.D. degree. His current research involves the application of evolutionary algorithms to heuristic search. Mr. Elyasaf won two Gold and one Bronze HUMIE awards (Human-Competitive Results Produced by Genetic and Evolutionary Computation) for his work.

Email: achiya.e@gmail.com

Recent years have seen a sharp increase in the application of evolutionary computation techniques within the domain of games. Situated at the forefront of this research tidal wave, Moshe Sipper and his group have produced a plethora of award-winning results, in numerous games of diverse natures, evidencing the success and efficiency of evolutionary algorithms in general—and genetic programming in particular—at producing top-notch, human-competitive game strategies. From classic chess and checkers, through simulated car racing and virtual warfare, to mind-bending puzzles, this book serves both as a tour de force of the research landscape and as a guide to the application of evolutionary computation within the domain of games.

An outstanding, timely book in the rapidly growing area of computational intelligence in games. A must read for both the neophyte and the seasoned researcher, with all the hallmarks of a landmark book.

John Koza, author of *Genetic Programming* tetralogy

In *Evolved to Win* Moshe Sipper provides a treasure trove of detailed examples and advice on using evolutionary computation, in conjunction with human expertise, to solve hard puzzles and to win a wide variety of challenging games. Sipper and his colleagues know this field better than anyone else, having produced some of the field's strongest and most exciting results, and this book provides a comprehensive tour of their results along with ample guidance for newcomers to the field.

Lee Spector, Professor of Computer Science, Hampshire College, &
Editor-in-Chief of the journal *Genetic Programming and Evolvable Machines*

Written by a top player in the field of computational intelligence in games, Moshe Sipper presents a cornucopia of games played—and won—by strategies attained through automatic programming. Whether you're a student or an experienced researcher, *Evolved to Win* is undoubtedly the book for you.

Natalio Krasnogor, Technical Editor-in-Chief of the journal *Memetic Computing*

News of the GP Bibliography

http://www.cs.bham.ac.uk/~wbl/biblio/

William B. Langdon — Computer Science, University College, London, UK (w.langdon@cs.ucl.ac.uk)

New features of the genetic programming bibliography include graphical displays of recent Internet based paper down load activity, html web pages identifying centres of GP expertise, new papers and a blog.

Essentially the genetic programming bibliography is a BibTEX file designed to support GP research [6; 3; 4]. Today, version 1.2568, it contains 9 585 GP papers. Since the first SIGEvolution article on the bibliography [2] in 2006 a number of additions have been made. These include: pages giving web usage and GP paper downloads (next section), links to other sources of information on genetic programming, links to active GP centres (Section 2), a page of new and modified BibTEX entries (Section 3), and my blog (Section 4).

1 Web Usage of the GP Bibliography

In 2006 Steve Gustafson added logging user browser activity with sitemeter. This provides an enormous range of data but perhaps the most interesting are the graphical displays. Figure 1 shows a fairly typical pattern of global use. Notice how user activity tends to align with the location of active GP authors (Section 2).

Although access to papers via the bibliography has been logged since 2006 [2], the graphical summaries are more recent. Figure 2 gives GP paper downloads for the past week. Figure 2 has a resolution of one hour, with the red line giving daily totals. Events such as the addition of papers from a major conference are often reflected by peaks in this graph.

The flags download graphic (Figure 3) is a more recent ambitious addition. The simple histogram in Figure 2 becomes a histogram made of small national flags. To be reasonably intelligible, even the miniaturised flags are much bigger than the corresponding unit area in Figure 2, consequently the time resolution allows only four bars per day rather than one per hour.

Figure 2 and Figure 3 cover the same period and so should contain the same data, albeit presented in different ways. However careful study will reveal differences that arise because the two scripts used to create them have different ways of trying to exclude web bot activity. (Recall from [2] that the vast majority of Internet traffic via the bibliography is machines talking to machines.) For example late in the night of Saturday June 14, 2014 there were many downloads from the same page in a few minutes which Figure 2 records as a spike but which Figure 3 regards as suspicious and ignores. In this case, Figure 2 is probably correct as the download corresponds to an Austrian user downloading each chapter of Pete Angeline's PhD thesis.

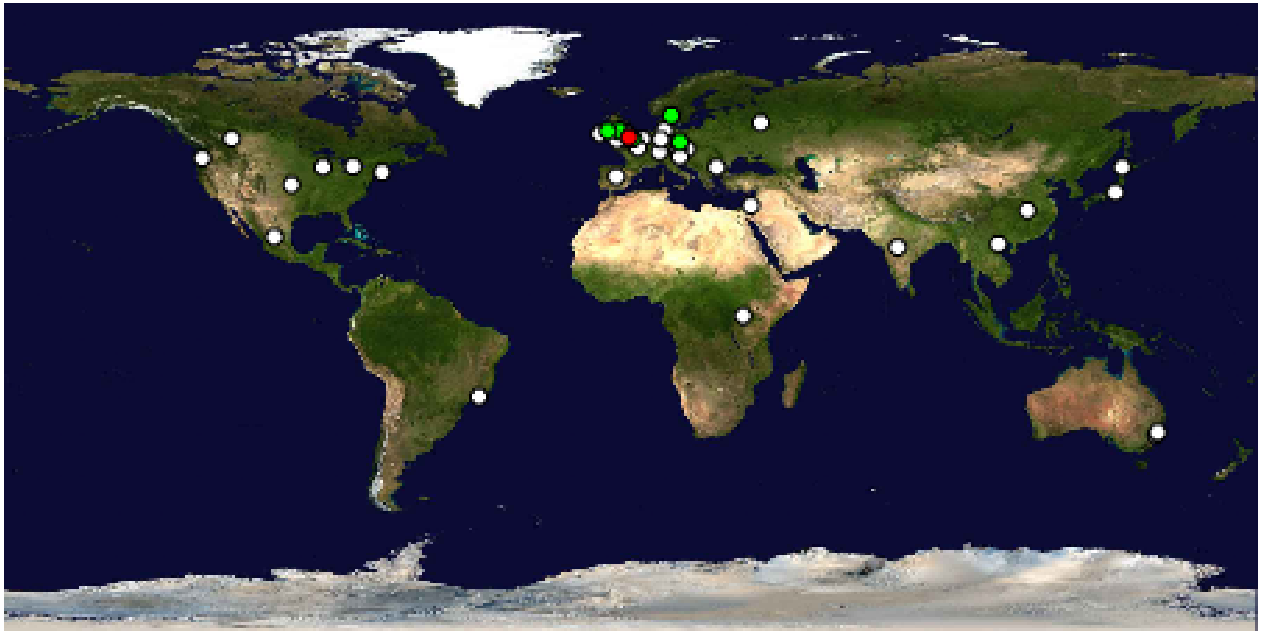


Fig. 1: 100 visitors to the genetic programming bibliography web site

1.1 Total GP Paper Downloads

The page about top papers contains an ordered list of the ≤ 100 most downloaded GP papers since 30 September 2006. Similarly, the page about top authors uses the same information to order GP paper downloads by author. Where a paper has multiple authors, a harmonic weighting scheme is used to divide the download unequally between its authors. Aniko Ekart suggested it should be documented how this is done. The scheme allocates twice as much weight to the first author as the second, three times as much as the third author, four times as much as the fourth and so on. Admittedly this is a somewhat arbitrary scheme but has the advantage each paper contributes the same amount regardless of the number of authors who wrote it and it is automatable. It gives a more even weighting than a proposed exponential scheme where each co-author would get twice as much credit as the next co-author. This harmonic scheme is the same scheme as is used to order authors in the main index page.

Every night the logs of user IP addresses are analysed to produce two ordered lists. In the first list, downloads are grouped by country requesting the paper. As expected, this approximately follows the locations of GP authors (see Section 2). In the second list, downloads are grouped by Internet domain. Surprisingly this list is dominated by commercial ISPs (e.g. comcast.net). University downloads (University College Dublin 201, Essex 199, and York 176) only appear after the first twenty commercial domains. Although it is tempting to ascribe the more than ten thousand paper downloads to non-academics interested in genetic programming, perhaps they are mostly due to academics working at home via commercially supplied networks.

2 Centres of Expertise

Since 2000 the bibliography has had associated with it a list of web pages on genetic programming. The vast bulk of home pages is links to authors of GP papers. Keeping all of these up to date has always proved a challenge.

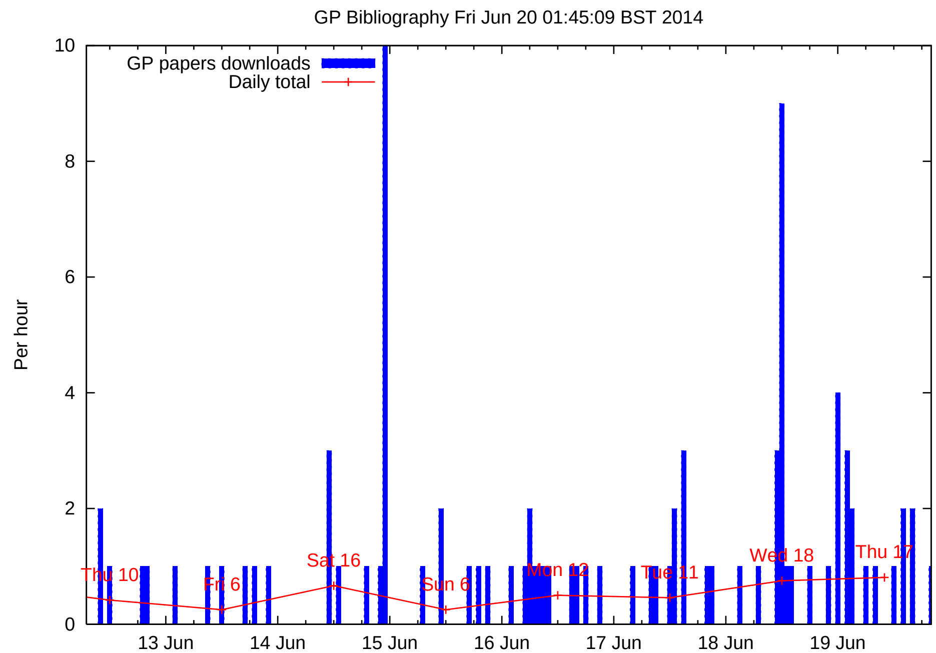


Fig. 2: Week of GP paper downloads via the genetic programming bibliography

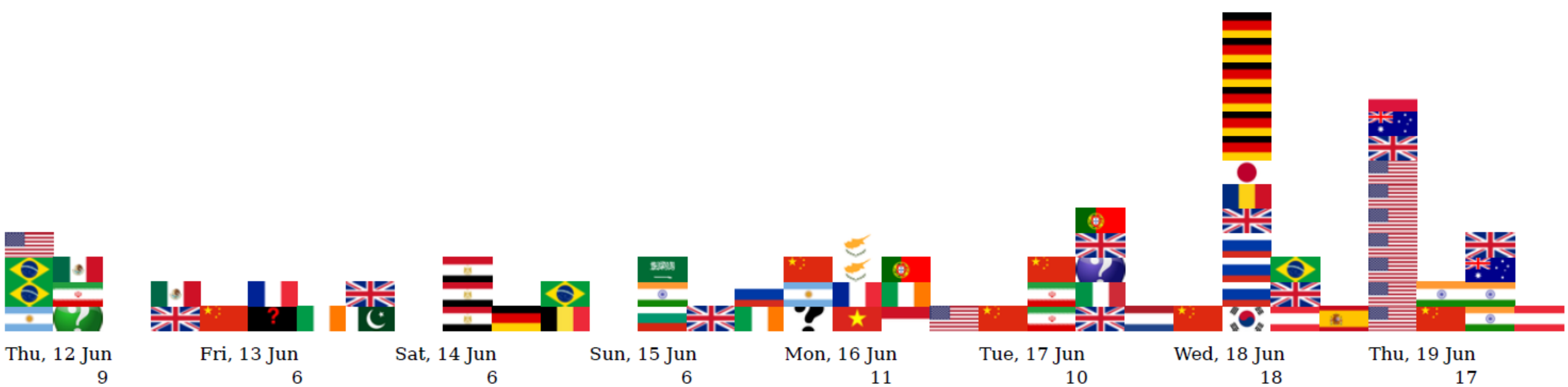


Fig. 3: A week of GP paper downloads via the genetic programming bibliography (excluding web bots). Each flag represents one GP paper download and contains the time of the download and a hyper text link to the paper within the bibliography. The flags are chosen based on each user's IPv4 address. The mapping from IPv4 to location is not perfect and where it cannot be inferred the flag is replaced by an unknown flag **?** per user.

In 2011 Adrian Carballal managed a dokuwiki based system which allowed users to both add and update the URL of their web home pages. However after six months of successful operation with many GPers actively participating, his dokuwiki was maliciously subjected to a cyber attack and eventually had to be withdrawn and so I had to revert to the original manual system.

At the start of each year inactive pages are expired. The original proposal was that if an author had not published a GP paper for more than five years, they would be deemed no longer active and their home page would be removed from the list. (Links to their home page from their papers in the genetic programming bibliography remain). However it was suggested that 5 years was too short and so the current period is ten years.

The 5 157 authors are divided by country. The division is not perfect as it is based on Internet domain names. However less than 2% of authors are classified as "unknown country". Some difficulties are encountered when authors put their home pages on a commercial site remote from their home institution. The URL of the commercial site may give the wrong location for the author. Also both commercial and academic sites may not release an author's URL when his home page moves elsewhere. This can defeat automated attempts to validate URLs.

Division by country is interesting in its own right. It comes as no surprise that the home of GP, the USA, has the most active GP authors (712). This has always been true. But in recent years China has usurped the UK (464) and is now in second place (497). To me it is surprising how well Spain (223) is doing, with many more active authors than the next European countries (Italy 159, Germany 126, France 101). In the Middle East, Iran (212) and India (213) have almost as many active GP authors as Spain. In the Far East, there is a gap between China 497 and Japan 266 and then a cluster: Taiwan 126, Australia 118, Korea 94 followed by New Zealand 62 and Singapore 53 and Malaysia 42. The big gap in the world's population of GP authors remains Africa with only three authors between the countries bordering the Mediterranean and South Africa. Another surprise is, given its size and educational traditions, the small number of known active GP authors in Russia (30). In some cases, e.g. Vietnam 7 and Ireland 92, GP success can be traced to individuals or universities.

Until recently the only sub-division attempted was by top-level domain name into countries. Recently Internet domains have been used to give a much finer hierarchical division. In many cases, authors have been grouped by university (e.g. UCL) and sometimes even into individual departments (e.g. CS.UCL). This has raised the tricky issue of how many people are sufficient to count as a "centre". Initially the threshold was ten but it was quickly reduced to five. Largely for the pragmatic reason that 5 names typically fit on a line. However even with a cluster size as small as five a large number of authors are in groups below the threshold and thus we get large amorphous "Others" at the end of the list for many countries. Given sufficient interest the raw data can be released to enable others to devise other presentation schemes.

3 New and Modified BibTEX Entries

The bibliography has now a web page listing both new entries and modified entries. This is created automatically by comparing the current and a previous version of the BibTEX file. Both lists are in alphabetic order by the first author. The modified list includes the number of lines of text that have been changed. As each release may have made very different numbers of changes, the page may cover more than just the last release. Instead it usually covers approximately up to the last month or so. Outside the bibliography various tools exist for automatically notifying you of changes. Also The Collection of Computer Science Bibliographies, which the GP bibliography feeds into, advertises an RSS feed for searches of the complete collection.

4 News

Started in 2012, the blog is simply a web page in reverse chronological order of minor events. It arose from a suggestion by Riccardo Poli, that each change should be announced to the community. So typically the start of the file will say a new release has been made and give the date when it was put on to The University of Birmingham web server. This will be followed by a list of recently added or updated genetic programming papers. These are represented by their BibTEX keys each of which contains a hypertext link directly to the entry in the GP bibliography for that paper. Sometimes the blog entry reports bugs or bug fixes.

Perhaps the most useful aspect of the blog is to give direct confirmation to authors that the change they requested has not only been done internally but that it is now included in the public release.

The source file `blog.src` is written as a single HTML table. The first column holds the date field. The second holds free form html text. blog.html is automatically generated overnight from `blog.src` by a Unix cron job in `php_crontab` using `blog.make`. Thus there is a delay of up to 24 hours between each new release appearing and blog.html being updated.

5 Who Cites GP Papers

A major missing part of the puzzle remains information on citation links to and between GP papers. The current hope is someone might construct a co-citation graph (similar to the existing co-authorship graph [7; 5]) by linking to or extracting data from existing tools, such as CiteSeer, or commercial tools, such as Google Scholar.

Acknowledgements

Since 1997, The University of Birmingham has continuously provided a home and support facilities to the bibliography. In all that time, outages have been measured in hours rather than days.

I started the bibliography in 1995 from data supplied by John Koza and published in [1, Appendix F]. Appendix F contains 100 papers (at the time Koza had an additional 49). Since [1] (published in 1994) the bibliography has been greatly extended. I would like to thank the many many GPers who have contributed references to it. Please continue.

References

[1] John R. Koza. *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press, Cambridge Massachusetts, May 1994.

[2] W. B. Langdon. Web usage of the GP bibliography. *SIGEvolution newsletter of the ACM Special Interest Group on Genetic and Evolutionary Computation*, 1(4):16–21, December 2006.

[3] W. B. Langdon and S. Gustafson. Genetic programming and evolvable machines: Five years of reviews. *Genetic Programming and Evolvable Machines*, 6(2):221–228, June 2005.

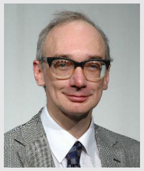
[4] W. B. Langdon and S. M. Gustafson. Genetic programming and evolvable machines: ten years of reviews. *Genetic Programming and Evolvable Machines*, 11(3/4):321–338, September 2010. Tenth Anniversary Issue: Progress in Genetic Programming and Evolvable Machines.

[5] W. B. Langdon, R. Poli, and W. Banzhaf. An eigen analysis of the GP community. *Genetic Programming and Evolvable Machines*, 9(3):171–182, September 2008.

[6] William B. Langdon. A bibliography for genetic programming. In Peter J. Angeline and K. E. Kinnear, Jr., editors, *Advances in Genetic Programming 2*, appendix B, pages 507–531. MIT Press, Cambridge, MA, USA, 1996.

[7] Marco Tomassini, L. Luthi, M. Giacobini, and W. B. Langdon. The structure of the genetic programming collaboration network. *Genetic Programming and Evolvable Machines*, 8(1):97–103, March 2007.

About the author



William B. Langdon is a principal research associate in UCL. He worked on distributed real time databases for control and monitoring of power stations at the Central Electricity Research Laboratories. He then joined Logica to work on distributed control of gas pipelines and later on computer and telecommunications networks. After returning to academe to gain a PhD in genetic programming at UCL (sponsored by National Grid plc.), he worked at the University of Birmingham, the CWI, UCL, Essex University, King's College, London and now for a third time at University College, London.

Homepage: http://www0.cs.ucl.ac.uk/staff/W.Langdon/
Email: w.langdon@cs.ucl.ac.uk

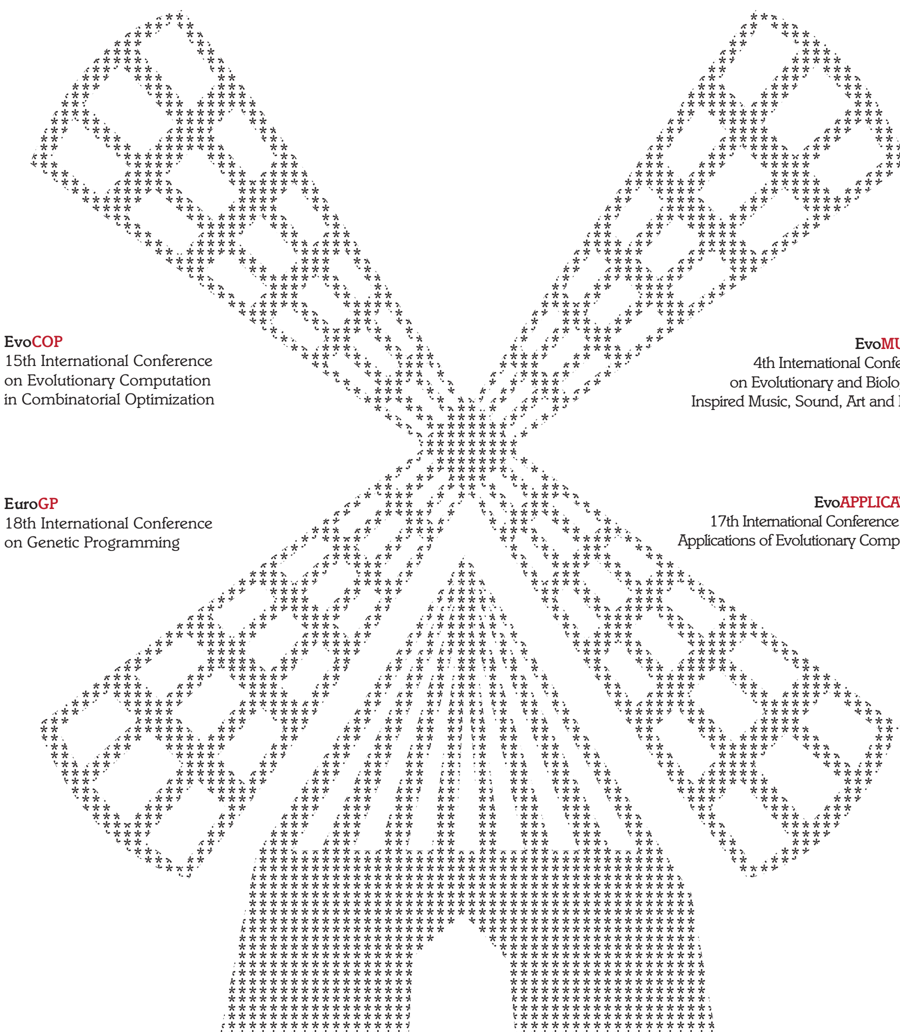
evo* 2015

8-10 April
Copenhagen - Denmark
www.evostar.org

Submission deadline
15 November 2014

EVO* coordinator
Jennifer Willies, Edinburgh Napier University, UK

Local Chairs
Paolo Burelli, Aalborg University
Julian Togelius, IT University of Copenhagen



EvoCOP
15th International Conference on Evolutionary Computation in Combinatorial Optimization

EvoMUSART
4th International Conference on Evolutionary and Biologically Inspired Music, Sound, Art and Design

EuroGP
18th International Conference on Genetic Programming

EvoAPPLICATIONS
17th International Conference on the Applications of Evolutionary Computation

Conferences Chairs: • Penousal Machado • Malcolm Heywood • James McDermott • Gabriela Ochoa • Francisco Chicano • Colin Johnson • Adrián Carballal • João Correia • Antonio Mora

Evo* Publicity Chairs: • Mauro Castelli • Pablo García Sánchez

EvoStar 2014 Event Report

Justyna Petke

In April Spain hosted five conferences: EuroGP, EvoCOP, EvoBIO, EvoMUSART and EvoApplications under the umbrella of EvoStar in the lovely city of Granada. The aim of the conferences is to bring together renowned researchers from all over the world using evolutionary computation and algorithms inspired by biology. Each conference focuses on particular applications of these methods. EvoStar history goes back to 1998, when EuroGP was first held in Paris, France. It has been held every year since and increasing number of conferences and workshops has been added over the years that led to establishing the EvoStar name in 2007. The 'Star' aims to mean all-inclusive, since $*$ can stand for anything in a regular expression.

Topics of presentations at EuroGP, which I attended, ranged from diagnosing breast cancer through learning differential equations using symbolic regression to automatically improving software.

The focus of EvoCOP is combinatorial optimisation, EvoBIO deals with biological matters, EvoMUSART targets applications of evolutionary algorithms in music, sound, art and design whilst EvoApplications contains 13 tracks on many diverse topics, including finance, communication networks and games.

This year's talks were enjoyed by participants from all over the world. EvoStar kicked off with the first plenary talk given by Professor Thomas Schmickl on 'Evolving bio-hybrid societies of animals and robots'. Prof. Schmickl has been experimenting with honeybees observing how they find an area with the best temperature (a global optimum) in different conditions. In particular, he showed that if just five bees out of 80 are trapped in a local optimum, then the majority will gravitate from the best point towards the five bees, even if they don't actually have any physical contact with them. These leader bees in human societies are known as 'early adopters'. He gave an example of people who buy Apple products as soon as they hit the market and then tell others via Facebook or Twitter to act similarly tempting them away from more economic solutions. It was a fascinating talk that showed how little we actually understand about the processes of what he called 'social cyborgs' like Google or Facebook.

During the second day the keynote speaker Professor Federico Moran, secretary general for Universities by the Spanish Ministry of Education, Culture and Sport, posted an intriguing question: 'what is life?'. He presented a journey from the origin of life to recent advances in molecular biology. He mentioned the work by Craig Venter who created a bacterial cell controlled by an entirely synthetic genome, and thus a question arose whether he actually created life. Prof. Moran claimed that this statement is false and that there is a lot more to life.

Aside from inspiring talks one could enjoy many wonderful sights that Granada has to offer. One could visit the famous Alhambra, the Granada Cathedral and walk around the lovely small alleys all around the city. Gala Dinner was held at the Carmen de los Mártires after an organised multilingual tour of the city. Conference organisation was generally well done, even though it would have been helpful to have a printed program.

Overall, the whole conference experience, starting from talks through venue to interesting conversations with conference participants, was very enjoyable and inspiring and would recommend attending it next year.

About the author

Justyna Petke is a research associate at the Centre for Research on Evolution, Search and Testing (CREST), located in the Department of Computer Science, University College London. She holds a BSc degree in Mathematics and Computer Science from University College London, and a DPhil in Computer Science from University of Oxford. Her current research interests include genetic improvement, combinatorial interaction testing as well as constraint solving.

Homepage: http://www0.cs.ucl.ac.uk/staff/J.Petke/
Email: j.petke@ucl.ac.uk

August 2014

IEEE Conference on Computational Intelligence and Games (CIG-2014)

August 26 - 29, 2014, Dortmund, Germany

Homepage: http://www.cig2014.de

Conference: August 26 - 29, 2014

Games can be used as a challenging scenery for benchmarking methods from computational intelligence since they provide dynamic and competitive elements that are germane to real-world problems. This conference brings together leading researchers and practitioners from academia and industry to discuss recent advances and explore future directions in this field.

The IEEE Conference on Computational Intelligence and Games is the premier annual event for researchers applying computational and artificial intelligence techniques to games. The domain of the conference includes all sorts of CI/AI applied to all sorts of games, including board games, video games and mathematical games. The yearly event series started in 2005 as symposium, and is a conference since 2009. An overview over the past CIG conferences is available at hrefhttp://www.ieee-cig.orgwww.ieee-cig.org, where you also find the proceedings. CIG 2014 will be hosted in the Park Inn hotel in the city center of Dortmund, a vibrant, technology-oriented city in the Ruhr area, Germany's largest metropolitan area with around 5 million people. The conference will consist of a single track of oral presentations, tutorial and workshop/special sessions, and live competitions. The proceedings will be placed in IEEE Xplore, and made freely available on the conference website after the conference.

Topics of interest include, but are not limited to:

- Learning in games
- Procedural content generation
- Player/opponent modeling in games
- Player affective modeling
- Player satisfaction and experience in games
- Computational and articial intelligence based game design
- Intelligent interactive narrative
- Theoretical or experimental analysis of CI techniques for games
- Non-player characters in games
- Comparative studies and game-based benchmarking
- Applications of game theory

General Chairs

Günter Rudolph, TU Dortmund, Germany

Mike Preuss, WWU Münster, Germany

Program Chairs

Mirjam Eladhari, University of Malta

Moshe Sipper, Ben-Gurion University of the Negev, Israel

Tutorials/Special Sessions Chair

Philip Hingston, Edith Cowan University, Perth, Australia

Competition Chair

Simon Lucas, University of Essex, UK

Keynote Chair

Gillian Smith, Northeastern University, Boston, USA

Proceedings Chair

Paolo Burelli, Aalborg University, Copenhagen, Denmark

September 2014

PPSN 2014 – International Conference on Parallel Problem Solving From Nature

September 13-17, 2014, Ljubljana, Slovenia

Homepage: http://ppsn2014.ijs.si

The 13th International Conference on Parallel Problem Solving from Nature (PPSN XIII) will be organized by the Jožef Stefan Institute, Ljubljana, Slovenia, and held at the Ljubljana Exhibition and Convention Centre on September 13-17, 2014. The conference aims to bring together researchers and practitioners in the field of Natural Computing. Natural Computing is the study of computational systems that use ideas and get inspiration from natural systems, including biological, ecological, physical, chemical, and social systems. It is a fast-growing interdisciplinary field in which a range of techniques and methods are studied for dealing with large, complex, and dynamic problems with various sources of potential uncertainties.

Paper Presentation Following the well-established tradition of PPSN conferences, all accepted papers will be presented during poster sessions. Each session will contain several papers, and will begin by a plenary quick overview of all papers in that session by a major researcher in the field. Past experiences have shown that such presentation format led to more interactions between participants and to deeper understanding of the papers.

General Chair

Bogdan Filipič, Jožef Stefan Institute, Slovenia

Honorary Chair

Hans-Paul Schwefel (Tech. Universität Dortmund, DE)

Program Co-Chairs

Thomas Bartz-Beielstein, Cologne University of Applied Sciences, DE
Jürgen Branke, University of Warwick, UK
Jim Smith, University of the West of England, UK

Tutorials Chairs

Shih-Hsi "Alex" Liu, California State University, Fresno, USA
Marjan Mernik, University of Maribor, Slovenia

Workshop Chairs

Evert Haasdijk, VU University Amsterdam, The Netherlands
Tea Tušar, Jožef Stefan Institute, Slovenia

Publication Chair

Jurij Šilc, Jožef Stefan Institute, Slovenia

Local Organizer

Gregor Papa, Jožef Stefan Institute, Slovenia

January 2015

Learning and Intelligent OptimizatioN Conference (LION9)

January 12-16, 2015, Lille, France

Submission deadline: October 10, 2014

Homepage: http://www.lifl.fr/LION9/

The large variety of heuristic algorithms for hard optimization problems raises numerous interesting and challenging issues. Practitioners are confronted with the burden of selecting the most appropriate method, in many cases through an expensive algorithm configuration and parameter tuning process, and subject to a steep learning curve. Scientists seek theoretical insights and demand a sound experimental methodology for evaluating algorithms and assessing strengths and weaknesses. A necessary prerequisite for this effort is a clear separation between the algorithm and the experimenter, who, in too many cases, is "in the loop" as a crucial intelligent learning component. Both issues are related to designing and engineering ways of "learning" about the performance of different techniques, and ways of using past experience about the algorithm behavior to improve performance in the future. Intelligent learning schemes for mining the knowledge obtained from different runs or during a single run can improve the algorithm development and design process and simplify the applications of high-performance optimization methods. Combinations of algorithms can further improve the robustness and performance of the individual components provided that sufficient knowledge of the relationship between problem instance characteristics and algorithm performance is obtained.

This meeting, which continues the successful series of LION events (see LION 5 in Rome, LION 6 in Paris, LION 7 in Catania, and LION 8 in Gainesville), is exploring the intersections and uncharted territories between machine learning, artificial intelligence, mathematical programming and algorithms for hard optimization problems. The main purpose of the event is to bring together experts from these areas to discuss new ideas and methods, challenges and opportunities in various application areas, general trends and specific developments.

Conference Organizers:
Clarisse Dhaenens
Laetitia Jourdan
Marie-Eléonore Marmion

Important Dates

Paper submission:	October 10, 2014
Author Notification:	November 25, 2014
Registration:	December 17, 2014
Camera ready:	January 3, 2015
Conference:	January 12-16, 2015

April 2015

Evostar 2015 - EuroGP, EvoCOP, EvoBIO and EvoWorkshops

April 8-10, 2015, Copenhagen, Denmark

Submission deadline: November 15, 2014

Homepage: www.evostar.org

EvoStar comprises of five co-located conferences run each spring at different locations throughout Europe. These events arose out of workshops originally developed by EvoNet, the Network of Excellence in Evolutionary Computing, established by the Information Societies Technology Programme of the European Commission, and they represent a continuity of research collaboration stretching back nearly 20 years.

The five conferences include:

- EuroGP 18th European Conference on Genetic Programming
- EvoBIO 12th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Computational Biology
- EvoCOP 15th European Conference on Evolutionary Computation in Combinatorial Optimisation
- EvoMUSART 4rd International Conference on Evolutionary and Biologically Inspired Music, Sound, Art and Design
- EvoApplications 16th European Conference on the Applications of Evolutionary and bio-inspired Computation including the following tracks
  - EvoCOMNET Application of Nature-inspired Techniques for Communication Networks and other Parallel and Distributed Systems
  - EvoCOMPLEX Applications of algorithms and complex systems
  - EvoENERGY Evolutionary Algorithms in Energy Applications
  - EvoFIN Track on Evolutionary Computation in Finance and Economics
  - EvoGAMES Bio-inspired Algorithms in Games
  - EvoHOT Bio-Inspired Heuristics for Design Automation
  - EvoIASP Evolutionary computation in image analysis, signal processing and pattern recognition
  - EvoINDUSTRY The application of Nature-Inspired Techniques in industrial settings
  - EvoNUM Bio-inspired algorithms for continuous parameter optimisation
  - EvoPAR Parallel and distributed Infrastructures
  - EvoRISK Computational Intelligence for Risk Management, Security and Defense Applications
  - EvoROBOT Evolutionary Computation in Robotics
  - EvoSTOC Evolutionary Algorithms in Stochastic and Dynamic Environments

Featuring the latest in theoretical and applied research, EVO* topics include recent genetic programming challenges, evolutionary and other meta-heuristic approaches for combinatorial optimisation, evolutionary algorithms, machine learning and data mining techniques in the biosciences, in numerical optimisation, in music and art domains, in image analysis and signal processing, in hardware optimisation and in a wide range of applications to scientific, industrial, financial and other real-world problems.

EVO* Poster

You can download the EVO* poster advertisement in PDF format here

EVO* Call for Papers

You can access the call for papers of all the EVO* conferences here.

EVO* Coordinator:

Jennifer Willies, Napier University, United Kingdom
j.willies@napier.ac.uk

General Chairs:

Penousal Machado, Malcom Heywood, James McDermott, Gabriela Ochoa, Francisco Chicano, Colin Johnson, Adrian Carballai, João Correia, Antonio Mora

Local Chair:

Paolo Burelli, Aalborg University
Julian Togelius, IT University of Copenhagen

Publicity Chair:

Mauro Castelli & Paolo García Sánchez

Important Dates

Submission Deadline:	15 November 2014
Notification:	07 January 2015
Camera-ready:	21 January 2015
Conference:	8-10 April 2015

May 2015

2015 IEEE Congress on Evolutionary Computation (CEC 2015)
May 25-28, 2015, Sendai, Japan
Homepage: http://sites.ieee.org/cec2015/
Deadline December 19, 2014

The annual IEEE CEC is one of the leading events in the field of evolutionary computation. It covers all topics in evolutionary computation including: Ant colony optimization, Artificial immune systems, Coevolutionary systems, Cultural algorithms, Differential evolution, Estimation of distribution algorithms, Evolutionary programming, Evolution strategies, Genetic algorithms, Genetic programming, Heuristics, metaheuristics and hyper-heuristics, Interactive evolutionary computation, Learning classifier systems, Memetic, multi-meme and hybrid algorithms, Molecular and quantum computing, Multi-objective evolutionary algorithms, Parallel and distributed algorithms, Particle swarm optimization, Theory and Implementation, Adaptive dynamic programming and reinforcement learning, Coevolution and collective behavior, Convergence, scalability and complexity analysis, Evolutionary computation theory, Representation and operators, Self-adaptation in evolutionary computation, Optimization, Numerical optimization, Discrete and combinatorial optimization, Multiobjective optimization.

IEEE CEC 2015 will feature a world-class conference that aims to bring together researchers and practitioners in the field of evolutionary computation and computational intelligence from all around the globe. Technical exchanges within the research community will encompass keynote lectures, regular and special sessions, tutorials, and competitions as well as poster presentations. In addition, participants will be treated to a series of social functions, receptions, and networking to establish new connections and foster everlasting friendship among fellow counterparts.

Important Dates:

- Competition Proposals Due: September 26, 2014
- Tutorial Proposals Due: January 9, 2015
- Special Session Proposals Due: October 31, 2014
- Paper Submission Due: December 19, 2014

More information can be found at: http://sites.ieee.org/cec2015/.

About the Newsletter

SIGEVOlution is the newsletter of SIGEVO, the ACM Special Interest Group on Genetic and Evolutionary Computation.

To join SIGEVO, please follow this link [WWW]

Contributing to SIGEVOlution

We solicit contributions in the following categories:

Art: Are you working with Evolutionary Art? We are always looking for nice evolutionary art for the cover page of the newsletter.

Short surveys and position papers: We invite short surveys and position papers in EC and EC related areas. We are also interested in applications of EC technologies that have solved interesting and important problems.

Software: Are you are a developer of an EC software and you wish to tell us about it? Then, send us a short summary or a short tutorial of your software.

Lost Gems: Did you read an interesting EC paper that, in your opinion, did not receive enough attention or should be rediscovered? Then send us a page about it.

Dissertations: We invite short summaries, around a page, of theses in EC-related areas that have been recently discussed and are available online.

Meetings Reports: Did you participate to an interesting EC-related event? Would you be willing to tell us about it? Then, send us a short summary, around half a page, about the event.

Forthcoming Events: If you have an EC event you wish to announce, this is the place.

News and Announcements: Is there anything you wish to announce? This is the place.

Letters: If you want to ask or to say something to SIGEVO members, please write us a letter!

Suggestions: If you have a suggestion about how to improve the newsletter, please send us an email.

Contributions will be reviewed by members of the newsletter board.

We accept contributions in LaTeX, MS Word, and plain text.

Enquiries about submissions and contributions can be emailed to editor@sigevolution.org.

All the issues of SIGEVOlution are also available online at www.sigevolution.org.

Notice to Contributing Authors to SIG Newsletters

By submitting your article for distribution in the Special Interest Group publication, you hereby grant to ACM the following non-exclusive, perpetual, worldwide rights:

- to publish in print on condition of acceptance by the editor
- to digitize and post your article in the electronic version of this publication
- to include the article in the ACM Digital Library
- to allow users to copy and distribute the article for noncommercial, educational or research purposes

However, as a contributing author, you retain copyright to your article and ACM will make every effort to refer requests for commercial use directly to you.