Metaheuristic Design Pattern: Interactive Solution Presentation

Mark R.N. Shackelford
Custom Software Development
Domestic and General Insurance
Brighton BN1 3FD
United Kingdom
mark.shackelford@domesticandgeneral.com

Christopher L. Simons
Department of Computer Science & Creative Technology
University of the West of England
Bristol BS16 1QY
United Kingdom
chris.simons@uwe.ac.uk

1. PROBLEM STATEMENT

In interactive metaheuristic search, the human helps to steer the trajectory of search by providing qualitative evaluations of solution individuals in the population. Given that much meta-heuristic search is typically population-based, it is challenging to design the presentation of solutions such that the human can provide effective qualitative evaluation [1]. Naively presenting each individual in a large population at each generation causes evaluation fatigue and a subsequent non-linearity of user focus making search trajectory inconsistent and ineffective [2].

Interactive search relies either on solely user-provided qualitative evaluation or a combination of user qualitative evaluation and machine-based quantitative fitness measures. User qualitative evaluation is often “multi-subjective” in that many fitness concerns are simultaneously evaluated [3]. Some evaluative concerns may be explicitly articulated by the user, although others are implicit. Thus the design of solution individual presentation is challenging but crucial to interactive metaheuristic search.

2. THE SOLUTION

Interactive Solution Presentation increases the focus and value of a single user interaction. A number of presentational mechanisms are available to achieve this, and these are grouped into potential design pattern abstractions:

2.1 Reduced Population Size

- *Rank-based presentation:* reduce population size significantly but present all candidate solutions in the population for rank ordering. User evaluation consists of placing candidate solutions in rank order rather than making absolute evaluations. Fitness values can be assigned to individuals in proportion to their rank.

2.2 Selected Individuals from Large Population

- *Banded presentation:* maintain a large population size but divide the population into N bands, where N is a suitably small number of solutions to present. One solution from each band is selected at random for presentation. The user evaluation of fitness for the band representative solution can be used to scale the original fitness values of all solutions in the band.
- *Partial sequential presentation:* maintain a large population size p, but select a reduced number for presentation at every p/N solutions from the population in fitness order, where N is suitably small number of solutions to present. Only the selected individuals have their fitness changed.
- *Partial random presentation*: maintain a large population size but select N solutions at random, where N is a suitably small number of solutions to present, and as with Partial Sequential presentation, only the selected individuals are changed.
- *Cluster representative presentation:* maintain a large population size but present a subset of the population to the user by eliminating solutions considered to be similar to their neighbors in the search space by means of ‘clustering’ algorithm based on some measure of distance between individuals. Those individuals presented to the user are thus considered to be representative of their close neighbors in the search landscape. Fitness values are assigned to individuals in proportion to their distance from the qualitatively evaluated individuals.
- *Surrogate presentation:* maintain a large population size but reduce the number of individuals presented to the user by applying ‘quick and dirty’ quantitative measures as surrogates of user evaluation to eliminate candidate solutions anticipated to be of little utility to the user.

2.3 Multiple Generations between Interactions

- *Fixed presentation interval:* present individuals from the population to users only after a fixed number of computational search iterations have elapsed rather than every iteration.
- *Fitness proportionate presentation interval:* present individuals to users at dynamic numbers of computational search iterations in proportion to overall population fitness. Where user evaluation is combined with computational quantitative calculation of fitness measures, early iterations of search rely on computational calculations. However, as iterations progress, the influence of user evaluation is more pronounced as population fitness increases.

3. CONSEQUENCES

For interactive search where qualitative user evaluation is the sole means of candidate solution fitness measurement, Reduced Population Size (e.g. *rank-based presentation)* can be beneficial. Conversely, for interactive search where qualitative user

GECCO '14, July 12 - 16 2014, Vancouver, BC, Canada

ACM 978-1-4503-2881-4/14/07…$15.00.
http://dx.doi.org/10.1145/2598394.2609849

evaluation is combined with computational quantitative fitness calculation, Selected Individuals (e.g. *banded, cluster representative* or *surrogate presentation)* can be valuable. For interactive search where large numbers of iterations are required for effective exploration of the search space, Multiple Generations (e.g. *fixed presentation interval* and *fitness proportionate presentation interval*) can beneficial.

Two competing forces drive a significant trade-off in interactive solution presentation. Firstly, it's important to minimize user evaluation fatigue, which requires that evaluation is effective in influencing the search trajectory to arrive swiftly at promising individuals and regions of search space. However, it's also important to maintain diversity in the population to ensure satisfactory exploration of the search space. *Banded*, *partial sequential, partial random, cluster representative* and *surrogate presentation* can all be useful when balancing this trade-off.

4. EXAMPLES

4.1 Rank-based Presentation

This has been widely used in a variety of interactive metaheuristic search applications. For instance, rank-based presentation has been used for large, real-world resource constrained multiple-project scheduling problems involving complex quality criteria [4]. Up to 25 evolving schedules are presented (as Gannt charts or Resource profiles) for ranking which enables the search to be guided by both the standard schedule quality criteria and also the master scheduler's non-formalized knowledge and experience. Rank-based presentation has also been used in the interactive evolution of user interfaces in the XUL interface definition language [5]. Rather than overload the user, the top fittest ten of a population of 100 interfaces are presented to the user, but the user has only to pick two for ranking. A further application of rank-based presentation can be seen in the adaptive surface inspection of hot freshly rolled steel sheets via interactive evolution [6]. An evolutionary strategy approach is used to generate eight offspring from a single parent individual image. When an engineer inspects the surfaces, images may be inspected side-by-side, and the eight individual images are ranked by the user scoring each image on a scale of 0 to 10. In a further different rank-based approach, the layout of a simple planar graph is evolved interactively by co-evolving a set of weights to take account of personal users' preferences based on the ranking of individual graph layouts [7].

4.2 Small Populations

An example of small populations of individuals presented to user for ranking relates to conceptual bridge design [8]. In this example, small populations (i.e. 20) of individual bridge designs comprising simple horizontal span bridges with and without support and angled span bridges with supports are presented to the user for ranking. Quantitative fitness measures relating to both the structural integrity and aesthetics of the bridge designs are combined with user-assigned ranking fitness to steer the search trajectory. In an example in the field of software engineering, rank-based presentation has been used in an interactive genetic algorithm for software re-modularization [9]. Two ranking presentation mechanisms are trialed. In the first mechanism, a number of individual software component pairs are presented to the user who simply ranks them as part of the same module or not. In the second, small modules containing a single software component are presented and the user is asked to rank other modules as the most appropriate to act as a placeholder for the component.

A further example of small populations of individuals being presented to users for ranked evaluation involves ergonomic chair design [10]. In this example, qualitative designer rankings of chair designs are combined with quantitative fitness measures relating to the structural integrity of chairs to steer the search trajectory. In a further example, rank-based presentation is also used with small population sizes for combining morphological fitness (e.g. fractal symmetries) with aesthetic fitness in jewelry design [11]. Indeed, small population sizes (i.e. < 10) have been recommended for interactive evolutionary computing to facilitate rank-based presentation [12].

4.3 Partial Presentation

Examples of *banded-presentation*, *partial sequential presentation* and *partial random presentation* can be found in interactive evolutionary search for drug discovery [13]. In this example, the goal is not to find an optimal solution or even a molecule that could be synthesized by a chemist, but rather to enable the user to explore the drug molecule design search space to arrive at fruitful candidate solutions that might be outside the user experience. The main intention with these methods is to maintain diversity (by keeping a large population) whilst at the same time allowing interactive fitness on a limited number of individuals.

4.4 Cluster Representative Presentation

This technique has been used in numerous engineering design domains where Cluster-Oriented Genetic Algorithms (COGAs) [14],[15] have been used in interactive search to identify high-performance regions of complex, multi-dimensional engineering design search spaces.

4.5 Surrogate Presentation

This interactive technique has been explored in a variety of search-based software testing systems where the user and computational search combine to generate effective test cases [16]. An 'Interaction Handler' component displays potential solutions to the user and collects feedback which is subsequently exploited to enable a dynamic preliminary selection of potential solutions for future presentation. In another example, decision maker preferences are exploited to enable strictly monotone progressively approximated value functions [17]. This preference-based approximated value function is used to select a subset of population individuals which are then evaluated by the decision maker in pair-wise comparisons. Surrogate presentation has also been used in the interactive search for early lifecycle software designs [18], where surrogate measures of software design elegance are used to select individuals from a large sized population for designer evaluation. In a further example of interactive genetic algorithm-based design of fashion garments, a surrogate model of user evaluation is used to alleviate user fatigue by building a classifier and a regressor to approximate the designer's cognition [19]. Two reliable training sets based on user evaluation are obtained, and then support vector classification and regression machines are trained as surrogate models. The input trained samples are the individuals evaluated by the user, and the output training samples of the classifier and the regressor are the widths and centers of individuals' fuzzy fitness assigned by the user, respectively.

4.6 Fixed Presentation Interval

Recommended by Kamalian et al. [20] as a concept to reduce user evaluation fatigue (in addition to reducing the number of individuals for user evaluation), fixed presentation interval has also been applied to interactive search-based software testing of embedded software systems [21].

4.7 Fitness Proportionate Presentation

This has been shown to be effective in interactive search for early lifecycle software designs [18], [22]. Quantitative computational calculations minimizing design coupling are combined with interactive user evaluation of design elegance. Initially, the number of iterations between user evaluations is high but as design coupling is minimized, the number of iterations between user evaluation decreases as qualitative assessments of design elegance increasingly steer search.

5. REFERENCES

[1] Tagaki, H. (2001) Interactive Evolutionary Computation: A Fusion of the Capabilities of EC Optimisation and Human Evaluation. *Proceedings of the ACM*. 78(9), 1275-1296.

[2] Shackelford, M.R.N. (2005) Implementation Issues for an Interactive Evolutionary Computation System. *Proceedings of the 2005 Genetic and Evolutionary Computation Conference (GECCO '05)*, 1045-1051, ACM Press.

[3] Miettinen, K.M. (1998) *Nonlinear Multiobjective Optimization.* Kluwer Academic Publishers.

[4] Shackelford M.R.N. and Corne, D. (2001) Collaborative Evolutionary Multi-project Resource Scheduling. *Proceedings of the 2001 Congress on Evolutionary Computing, vol. 2,* 1131-1138, IEEE Press.

[5] Quiroz, J.C., Louis, S.J. and Dascalu, S.M. (2007) Interactive Evolution of XUL User Interfaces. *Proceedings of the 2007 Annual Conference on Genetic and Evolutionary Computation (GECCO '07)*, 2151-2158, ACM Press.

[6] Caleb-Solly, P. and Smith, J.E. (2007) Adaptive Surface Inspection via Interactive Evolution. *Image and Vision Computing*, 25(7), 1058-1072.

[7] Barbosa, H.J. and Barreto, A.M. (2001) An Interactive Genetic Algorithm with Co-evolution of Weights for Multi-objective Problems. *Proceedings of the 2001 Annual Conference on Genetic and Evolutionary Computing (GECCO '01)*, 203-210, Morgan Kaufman.

[8] Machwe, A.T. and Parmee, I.C. (2007) Multi-objective Analysis of a Component-based Representation within an Interactive Evolutionary Design System. *Engineering Optimisation.* 39(5), 591-613.

[9] Bavota, G., Carnevale, F., De Lucia, A., Di Penta, M. and Oliveto, R. (2012) Putting the Developer in-the-loop: An Interactive GA for Software Re-modularization. *Proceedings of the 4th International Symposium on Search-Based Software Engineering (SSBSE 2012), LNCS 7515,* 75-89, Springer-Verlag, Heidelburg, Berlin.

[10] Brintrup, A.M., Ramsden, J., Takagi, H. and Tiwiri, A. (2008) Ergonomic Chair Design by Fusing Qualitative and Quantitative Criteria Using Interactive Genetic Algorithms. *IEEE Transactions on Evolutionary Computation.* 12(3), 343-354.

[11] Wannarumon. S., Bohez, E.L.J. and Annanon, K. (2008) Aesthetic Evolutionary Algorithm for Fractal-based User-cenered Jewely Design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing.* 22(1), 19-39.

[12] Bentley, P. (1999) An Introduction to Evolutionary Design by Computers. In *Evolutionary Design by Computer* (Bentley, P. Ed.), 1-73, Morgan Kaufmann.

[13] Shackelford, M.R.N. (2014) Evolutionary Algorithm for Drug Discovery Interim Design Report. Online: http://arxov.org/abs/1403.4871

[14] Parmee, I.C., Cvetkovic, D, Watson, A.H. and Bonham, C.R. (2000) Multiobjective Satisfaction within an Interactive Evolutionary Design Environment. *Evolutionary Computation.* 8(2), 197-222.

[15] Bonham, C.R. and Parmee, I.C. (2004) Developments of the Cluster-Oriented Genetic Algorithm (COGA). *Engineering Optimisation.* 36(2), 249-279.

[16] Marculescu, B., Feldt R. and Torkar, R. (2012) A Concept for an Interactive Search-Based Software Testing System. *Proceedings of the 4th International Symposium on Search-Based Software Engineering (SSBSE 2012), LNCS 7515,* 273-278, Springer-Verlag, Heidelburg, Berlin.

[17] Deb, K., Sinha, A., Korhonen, P.J and Wallenius, J. (2010) An Interactive Evolutionary Multiobjective Optimization Method Based on Progressively Approximated Value Functions*. IEEE Transactions on Evolutionary Computation*. 14(5), 723-739.

[18] Simons, C.L. and Parmee, I.C. (2012) Elegant Object-oriented Software Design via Interactive Evolutionary Computation. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews.* 42(6), 1797-1805.

[19] Sun, X, Gong, D. and Li, S. (2009) Classification and Regression-based Surrogate Model-assisted Interactive Genetic Algorithm with Individual Fuzzy Fitness. *Proceedings of the 2009 Annual Conference on Genetic and Evolutionary Computation (GECCO '09)*, 907-914, ACM Press.

[20] Kamalian, R., Yeh, E., Zhang, Y., Agogino, A. and Takagi, H. (2006) Reducing Human Fatigue in Interactive Evolutionary Computation through Fuzzy Systems and Machine Learning Systems. *Proceedings of the 2006 IEEE International Conference on Fuzzy Systems*, 678-684, IEEE Press.

[21] Marculescu, B., Feldt R. and Torkar, R. (2013) Objective Re-Weighting to Guide an Interactive Search-Based Software Testing System. *Proceedings on the 12th International Conference on Machine Learning and Applications*, in press, IEEE Press.

[22] Simons, C.L., Smith, J.E. and White, P. (2012) Interactive Ant Colony Optimization (iACO) for Early Lifecycle Software Design. Online: http://arxiv.org/abs/1212.5461