AUTOMATIC PARAMETERIZATION FOR EXPEDITIOUS MODELLING OF VIRTUAL URBAN ENVIRONMENTS: A NEW HYBRID META HEURISTIC

Filipe Cruz, António Coelho
FEUP - Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
filipe.cruz@fe.up.pt, acoelho@fe.up.pt

Luis Paulo Reis
FEUP - Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
LIACC - Laboratório de Inteligência Artificial e Ciência de Computadores, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
lpreis@fe.up.pt

Keywords: Metaheuristics, Optimization Algorithms, Optimization Systems, Expeditious Modelling.

Abstract: Expeditious modelling of virtual urban environments consists of generating realistic 3d models from limited information. It has several practical applications but typically suffers from a lack of accuracy in the parameter values that feed the modeller. By gathering small amounts of information about certain key urban areas, it becomes possible to feed a system that automatically compares and adjusts the input parameter values of modelling rules to find optimal parameter combinations. These optimum parameterized rules can then be reapplied in the generation of virtual urban models of areas with missing data but similar characteristics to the referenced area. Based on several nature inspired meta heuristic algorithms such as genetic algorithms, simulated annealing and harmony search, this paper presents a new hybrid meta heuristic algorithm capable of optimizing functions with both discrete and continuous parameters and offer competitive results in a highly neglected field of application.

1 INTRODUCTION

There has been a growing need for expeditious modelling systems for urban environments in recent years. Applications include virtual city tours, georeferenced services, urban planning.

Creating urban models that are accurate representations of the real world can be extremely hard in terms of information gathering and invested man power. Close resemblances to the real world is often an acceptable compromise for some applications. It often occurs that it is quite impossible to gather information from the entire city you are trying to model but it is somewhat easy to extract detailed information from certain key-defining small areas from within the city. Generating expeditious modelling rules that will match such information is an important requirement. The artificial intelligence field of optimum search includes meta heuristic algorithms that can assist in determining optimum parameter values for modelling rules.

The work described in this paper focuses on solving this problem by developing a new hybrid meta heuristic algorithm capable of handling both discrete and continuous parameters with competitive performance by adapting ideas from several other nature based meta heuristic algorithms such as genetic algorithms, simulated annealing and harmony search.

This paper is divided in five sections: the first serving as the introduction, the second section referring to related work, the third describing the developed system, the fourth presenting and discussing the results achieved, and the fifth presenting the conclusions and future work.

2 RELATED WORK

2.1 Expeditious Modelling of Virtual Urban Environments

The modelling of virtual urban environments has many different applications including virtual city tours (Schilling and Coors, 2003), georeferenced services (Ito et al., 2005), cultural heritage preservation (Hildebrand et al., 2000) (Zach et al., 2001) and urban planning. Often there is a need for realistic or semirealistic models of cities, however, modelling accurate realistic models grows problematic considering increasing needs in size and complexity.

Using principles of l-system (Lindenmayer, 1968) (Prusinkiewicz, 1986), it is possible to define sets of production rules to model all kinds of urban environment elements (Parish and Muller, 2001) based on limited georeferenced data (Muller et al., 2006). It's often difficult to acquire reliable data of an entire city we are trying to model, however it is relatively easy to gather information from certain key-defining sections of a city and assign the data to a georeferenced database. The production rules with the data collected will enable the creation of models resembling the area of gathered information.

2.2 Metaheuristic Algorithms for Solving Optimization Problems

The field of artificial intelligence branches many areas (Russel and Norvig, 2003), one of the most relevant ones is the field of optimum search. Meta heuristic algorithms play an important role in this field. Different meta heuristic variants and hybrid versions refer to evolutionary computation or nature inspired behaviours: genetic algorithms (Holland, 1992) (Goldberg, 2002), simulated annealing (Kirkpatrick et al., 1983) (Aarts et al., 2005) and harmony search (Geem et al., 2001) (Lee and Geem, 2005) (Mahdavi et al., 2007).

The common principle is to find the best combination of parameters of a given vector function such that a related objective function is maximized or minimized. This process is iterated through educated random guesses following the heuristic logic of the algorithm. There is no best heuristic, performance varies according to the constraints of problem. A big number of proposals for hybrid or customized adaptations is found in recent literature (Deep and Thakur, 2007) (Arumugama et al., 2005) (Kumar et al., 2007) (Lee and Geem, 2005) (Mishra et al., 2005).

3 IMPLEMENTATION

3.1 Problem Statement

An l-systems based expeditious modeller for the generation of realistic urban environments becomes more valuable with an optimum parameterization system.

The system must handle the input of boolean, integer and real based parameters. The system must allow a configuration easily adaptable to the problem. When solving linear constrained problems the system must out-perform simple random based algorithms such as hillclimber and random search. When solving problems with multiple local maximums the system is required to perform above par of classic meta heuristic algorithms such as simulated annealing, genetic algorithm and harmony search.

3.2 Hybrid Optimizer Metaheuristic Algorithm

The hybrid meta heuristic is inspired by basic principles of real based genetic algorithms (Michalewicz, 1994) (Davis, 2001) and concepts from simulated annealing and harmony search. Genetic Algorithms typically require big populations and many parameter configurations to choose the best parent selection, cross over method and mutation operands for the problem at hand. Simulated annealing flaws due to it's limited local search nature. Harmony Search can be considered as a variation of standard genetic algorithms, similarly suffering from some of it's flaws. Hybrid implementations of genetic algorithms or harmony search with new operands that somewhat enable a simulated annealing inspired behaviour have been documented in the literature before. Our version pushes that envelope further in a fully configurable implementation connected to a previously developed expeditious urban modelling system.

The optimizer module is structured as follows: there are two families of populations resident in memory at all times, the original parent family and the top list family. Their dimensions can be configured by XML. Each iteration step of the metaheuristic algorithm consists on creating a new original parent family generation. The top list family maintains in memory the best solutions ever found so far, sorted by quality.

Each solution stores values for all parameters being calibrated. Each parameter has information regarding its type (integer, boolean, real) and scope (minimum and maximum values). The type and scope for each parameter are pre-configured by XML. The values for the first generation of the original parent family are calculated randomly within its scope boundaries. The first generation of the top list family is obtained by sorting the first generation of original parent family. These values can also be loaded from disk to test scenarios in same starting terms.

Each following original parent generation is obtained by cross-breeding the original parent family with a chosen member of the top list family. Ensuring an elitist selection behaviour inspired by genetic engineering. Several threshold variables further influence the selection of the new solution to ensure a wider search space scope not limited to the first generation. These variables incorporate monte carlo methods (Metropolis and Ulam, 1949) using probability thresholds inspired by simulated annealing and harmony search. There are a total of five threshold parameters which must be calibrated considering the problem.

Some of these thresholds are, or can, be affected by an internal value, refered to as *globalentropy* which increases by each passing generation as described in (1).

$$globalentropy = iterationstep/maxsteps \quad (1)$$

Random New Struct Threshold (*trns*), affects the probability of choosing a completely random new solution. The higher this value the more probable it becomes to occur a total random creation of a new solution as seen in formula (3). Harmony Search as a similar probabilistic threshold to ocasionaly select entirely new solutions.

$$trns = thresholdRandomNewStruct \quad (2)$$

$$randSol = rand() * globalentropy \ < \ trns \quad (3)$$

Random New Type Threshold (*trnt*), affects the probability of choosing a completely random new value for each of the solution parameter types. The higher this value is the more probable it is to occur a totally random new value for the current parameter type of the solution as seen in formula (5).

$$trnt = thresholdRandomNewType \quad (4)$$

$$randType = rand() * globalentropy \ < \ trnt \quad (5)$$

Toplist Dispersion Threshold (*ttld*), affects the probability of choosing lower ranking *toplist* parents to cross the solution with. The higher this value the wider the scope of choice as seen in formula (8). This formula can be somewhat comparable to the roulette wheel parent selection method of real based genetic algorithms (Michalewicz, 1994).

$$ttld = thresholdToplistDispersion \quad (6)$$

$$ttld > 1.0 \ : \ ttld = 1 - globalentropy \quad (7)$$

$$victim = (rand() * ttld * maxFamilySize) \quad (8)$$

Typevalue Dispersion Threshold (*ttvd*), affects the parental gene influence for each value of the parameter types of the solution as seen in formula (13). This formula can be somewhat comparable to a hybrid version of the real based genetic algorithms operands of average convex cross over and direction based cross over (Davis, 2001).

$$tlv = topslistParentValue \quad (9)$$

$$orv = originalParentValue \quad (10)$$

$$ttvd = thresholdTypevalueDispersion \quad (11)$$

$$ttvd > 1.0 \ : \ ttvd = 1 - globalentropy \quad (12)$$

$$newvalue = (ttvd * orv) + ((1 - ttvd) * tlv) \quad (13)$$

Typevalue Entropy Threshold (*ttve*), affects the probability of scope jitter for each value of the parameter types of the solutions can be seen in formula (17). This threshold can be compared to the *pitch* adjustment of harmony search and several recent real based genetic algorithm mutation operands described in the literature.

$$ttve = thresholdTypevalueEntropy \quad (14)$$

$$ttve > 1.0 \ : \ ttve = 1 - globalentropy \quad (15)$$

$$scope = \|(OParentValue - TLParentValue)\| \quad (16)$$

$$igl = (1.0F - globalentropy) \quad (17)$$

$$ttvss = igl * igl * igl \quad (18)$$

$$range = MaxParamValue - MinParamValue \quad (19)$$

$$scope < ttvss * range \ : \ maxscope = range * igl \quad (20)$$

$$scope > ttvss * range \ : \ maxscope = scope * ttve \quad (21)$$

$$maxscope = scope * rand() \quad (22)$$

$$newvalue = newvalue + scope - (maxscope/2) \quad (23)$$

4 RESULTS

A test-case was prepared involving the parameterization of a set of production rules which would model several buildings with certain height values missing. The known information from all of the buildings included georeferenced location and the values of the buildings perimeter, area and *bottomzvalue*. The unknown information from some of the buildings comprised solely the *topzvalue*.

The production rule used to estimate the unknown *topzvalue* from the buildings is described mathematicaly in formula (27).

$$av = (cra - 1) * fra * area \quad (24)$$

$$ab = (crb - 1) * frb * bottomzvalue \quad (25)$$

$$ap = (crp - 1) * frp * perimeter \quad (26)$$

$$topzvalue = avgz + disp * (av + ab + ap) \quad (27)$$

The formula implies a relation between the building's area (24), perimeter (26) and *bottomzvalue* (25) with the building's height to estimate a realistic *topzvalue* for input to the expedite modeler.

Our formula has a total of 8 unknown fields to be parametrized: *avgz* [100.0 .. 120.0], the average height for all the buildings. *disp* [0.0 .. 1.0], the dispersion rate from the average height. *cra* [0 .. 3], area correlation. *crb* [0 .. 3], *bottomzvalue* correlation. *crp* [0 .. 3], perimeter correlation. *fra* [0.0 .. 1.0], the area value correlation factor. *frb* [0.0 .. 1.0], the *bottomzvalue* correlation factor. *frp* [0.0 .. 1.0], the factor of the perimeter correlation.

Different configurations of the metaheuristic algorithm were tested with a fixed *toplist* family size of 20. The different tested configurations include the behaviour of some classic algorithms: random search, hill climber and simulated annealing. A few additional configurations were also tested for comparative performance results. Each test iterated 50 generations with a family size of 8 and were labbeled as follows: *rnd*, random search. *h1n*, hybrid new configuration similar to standard harmony search. *h2n*, hybrid second new configuration similar to standard genetic algorithms. *h3n*, hybrid third new configuration similar to an over elitist version of genetic algorithms. *hhc*, hybrid hill climber performing as a standard implementation. *hsa*, hybrid simulated annealing performing as a standard implementation.

The calibration parameters of each configuration tested can be consulted in Table 1.

Table 1: Threshold parameters of the different configurations.

config	trns	trnt	ttld	ttvd	ttve
rnd	1.0	1.0	0.0	0.5	0.5
h1n	0.01	0.01	0.1	1.1	1.1
h2n	0.001	0.001	0.4	0.15	1.1
h3n	0.001	0.001	0.1	0.1	1.1
hhc	0.01	0.01	0.0	1.0	0.1
hsa	0.01	0.01	0.0	1.0	1.1

All simulations were performed three times to present some insight on how deeply the performance of the meta heuristics algorithm stochastic nature is affected.

The quality function for the test case is calculated as a weighted sum of the height from the involved buildings.

Table 2 displays the progressive results obtained from our testcase. It shows the quality value of the best solution from the *toplist* at 2%, 40%, 80% and 100% of the iteration process. The results in Table 2 below helps identify how fast a certain configuration has progressed and the degree of results dependancy on the initial randomized family generation.

5 CONCLUSIONS AND FUTURE WORK

An automatic parameterization system for expeditious modelling of virtual urban environments has been successfully developed.

Our test case, despite its relatively low complexity, successfully demonstrated the potential use of the our new hybrid meta heuristic algorithm in facilitating expeditious modelling scenarios.

Table 2: Progressive results of the different configurations.

config	2%	40%	80%	100%
rnd-1	163.95	54.254	44.854	44.854
rnd-2	373.90	98.338	14.557	14.557
h1n-1	272.53	9.319	9.319	9.319
h1n-2	29.479	10.910	8.972	8.972
h2n-1	438.53	24.047	14.425	14.425
h2n-2	1038.61	1.758	1.758	1.758
h3n-1	22.164	6.418	0.175	0.175
h3n-2	288.56	3.934	0.798	0.798
hhc-1	117.35	47.728	47.728	32.145
hhc-2	1069.77	427.75	232.77	104.08
hsa-1	1207.92	75.354	40.354	40.354
hsa-2	315.95	137.05	42.689	9.804

Further tests are required to statistically compare the performance of the new hybrid meta heuristic algorithm with other meta heuristic algorithms and parameter optimization problems.

An envisioned improvement to the system involves applying principles of nested partition and linear regression to strengthen performance.

REFERENCES

Aarts, E., Korst, J., and Michiels, W. (2005). Simulated annealing. In *In Edmund K. Burke and Graham Kendall, editors, Search Methodologies*, pages 187–210. Springer.

Arumugama, M., Raoa, M., and Palaniappanb, R. (2005). New hybrid genetic operators for real coded genetic algorithm to compute optimal control of a class of hybrid systems. In *Applied Soft Computing 6*, pages 38–52.

Davis, L. (2001). *Handbook of Genetic Algorithms*. Van Nostrand Reinhold.

Deep, K. and Thakur, M. (2007). A new mutation operator for real coded genetic algorithms. In *Appl. Math. Comput. (2007).*

Geem, Z., Kim, J., and Loganathan, G. (2001). A new heuristic optimization algorithm: Harmony search. In *SIMULATION, Vol. 76, No. 2*, pages 60–68.

Goldberg, D. (2002). *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Addison-Wesley, Reading, MA.

Hildebrand, A., Dahne, P., Seibert, F., Christou, I., Demiris, A., Diorinos, M., Ioannidis, N., Almeida, L., Diogo, A., and Weidenhausen, J. (2000). Augmented reality based system for personalized tours in cultural heritage sites. In *International Conference of Augmented Reality, Virtual Environments and 3D Imaging Proceedings, Grcia.*

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA.

Ito, H., Teh, S., Nakanishi, H., and Tagawa, T. (2005). Design and implementation of 3d interface for digital city. In *Electronics and Communications in Japan (Part III: Fundamental Electronic Science), n. 6*, volume 88, pages 60–68.

Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). Optimizing by simulated annealing. In *Science, Number 4598*, volume 220.

Kumar, A., Sharma, D., and Kalyanmoy, D. (2007). A hybrid multi-objective optimization procedure using pcx based nsga-ii and sequential quadratic programming. In *KanGAL Report Number 2007008*.

Lee, K. and Geem, Z. (2005). A new meta-heuristic algorithm for continuous engineering optimization harmony search theory and practice. In *Computational Methods Applied Mechanical Engineering*, volume 194, pages 3902–3933.

Lindenmayer, A. (1968). Mathematical models for cellular interaction in development, parts i and ii. In *Journal of Theoretical Biology*, volume 18, pages 280–315.

Mahdavi, M., Fesanghary, M., and Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. In *Applied Mathematics and Computation 188*, pages 1567–1579.

Metropolis, N. and Ulam, S. (1949). The monte carlo method. In *Journal of the American Statistical Association, number 247*, volume 44, pages 335–341.

Michalewicz, Z. (1994). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag.

Mishra, N., Prakash, M., Tiwari, R., Shankar, F., and Chan, T. (2005). Hybrid tabusimulated annealing based approach to solve multi-constraint product mix decision problem. In *Expert Systems with Applications 29*.

Muller, P., Wonka, P., Haegler, S., Ulmer, A., and Gool, L. (2006). Procedural modeling of buildings. 25(3):614–623.

Parish, Y. and Muller, P. (2001). Procedural modeling of cities. In Fiume, E., editor, *Proceedings of ACM SIGGRAPH 2001*, pages 301–308, New York, NY, USA. ACM Press.

Prusinkiewicz, P. (1986). Graphical applications of l-systems. In *ACM Computer Graphics (Proceedings of Siggraph86 Vision Interface86), CIPS*, pages 247–253.

Russel, S. and Norvig, P. (2003). *Artificial Intelligence: A modern approach*. Prentice-Hall, 2 edition.

Schilling, A. and Coors, V. (2003). Generation of vrml city models for focus based tour animations. In *Proceeding of the Eighth International Conference on 3D Web Technology, Web3D 2003, Saint Malo, France*, pages 39–48.

Zach, C., Klaus, A., Bauer, J., Konrad, K., and Grabner, M. (2001). Modeling and visualizing the cultural heritage data set of graz. In *Proceedings of the 2001 Conference on Virtual Reality, Archeology, and Culture Heritage*, pages 219–226.