Hybrid Heuristic Methodologies applied in the Parameterization of Virtual Urban Environments

Filipe Cruz[1], António Coelho[1], and Luis Paulo Reis[1 2]
filipe.cruz@fe.up.pt acoelho@fe.up.pt lpreis@fe.up.pt

[1] DEI/FEUP - Departamento de Engenharia Informática, Faculdade de Engenharia da Universidade do Porto
[2] LIACC - Laboratório de Inteligência Artificial e Ciência de Computadores da Universidade do Porto

Abstract. Expeditious modeling of virtual urban environments has many applications but typically suffers from requiring intensive trial and error to determine reliable generation rules or lacking complete data to input the procedural modeling. Using optimization algorithms it is possible to automate the parametrization of production rules and estimate missing data. This paper presents a hybrid heuristic optimization algorithm developed to solve parameterization problems with mixed discrete and continuous parameter types as a viable solution in a neglected field of application, presenting a successful application in a real scenario and competitive results in performance tests under a standard batch of non-convex non-linear continuous parameter functions.

1 INTRODUCTION

The modeling of virtual urban environments has many different applications including virtual city tours [1], location based services [2], cultural heritage preservation [3] [4] and urban planning. There is a high demand for realistic or semi-realistic models of cities, however, modeling these environments using standard 3d modeling tools by hand grows problematic in terms of required expertise and available time. Expeditious modeling solutions have grown in demand [5] to automate or simplify some of the complex modeling process, either by acquiring and interpreting graphical information [6] [7] [8] or by the use of production rules to generate procedural models [9] based on limited geographically referenced data. Certain expeditious urban modeling systems use adapted L-System [10] [11] principles to define sets of production rules capable of modeling all kinds of urban environment elements [12].

The procedural modeling rules used in these systems are often complex, fine tuning the parameters can become very difficult even with domain knowledge available. In this regard, the artificial intelligence fields of stochastic optimization and evolutionary computation can greatly assist to develop or embed systems capable of determining the optimum parameter combination for different types of combinatorial problems, improving expeditious modeling systems to reach a higher standard of automation, reliability and realism.

As an example, it often occurs that it is impossible, very troublesome or highly cost-ineffective to acquire complete accurate data for an entire city we are trying to model.

However it is often possible, and relatively accessible to gather information from certain key-defining areas within the city and assign the data to a geographically referenced database. Production rules can be optimized and calibrated with the data collected to generate models resembling this area of gathered information. Such production rules can then be reapplied on areas of similar characteristics or more limited information in order to estimate data that generates more realistic models.

The purposed methodology is the development of a hybrid meta heuristic algorithm capable of calibrating parameters of typical procedural modeling rules. Combinatorial problems in this specific area can vary significantly in type and complexity, in this sense a hybrid and reconfigurable system was considered to be a more logical approach to ensure the system could be adapted to solve any type of black box problem with parameter inputs of discrete and continuous nature alike. Preliminary work on this methodology by the authors has been presented at the International Conference on Informatics, Control, Automation and Robotics 2008 [13][3] and might be worth consulting.

This paper is divided in four sections: the first serving as an introduction, the second describing the theory behind the developed system, its architecture and performance comparison. The third section presenting an application of the system in the real world. The fourth some conclusions on the work and ideas for future development.

2 HYBRID HEURISTIC OPTIMIZATION ALGORITHM

Meta heuristic stochastic optimization algorithms [14] can be somewhat generically divided by some authors in two generic families: local search (LS) and population based (PB). LS algorithms such as random search, hill climbing [15], tabu search [16] and simulated annealing (SA) [17] [18], typically focus on analysing a set of constrained neighborhood solutions iteratively gathered around the best solution found so far, the iterative process repeats until a certain criteria is met. PB algorithms such as real coded genetic algorithms (RCGA) [19] [20] [21], harmony search (HS) [22] [23], particle swarm [24] and ant colony based [25] optimization all typically function by keeping in memory a family of solutions and iteratively creating new generations to be analysed. Most LS and PB methods can outperform each other given a specific type of problem. A careful balance is required in selecting the most fitting algorithm and its optimum configuration to match the problem at hand to avoid common pitfalls in determining the global optimum with the least amount of required iterations.

More recent approaches from evolutionary computing [26] present optimization algorithms focused on self-adaptation with acclaimed results such as the Covariance Matrix Adaptation Evolution Strategy (CMA ES) introduced by Nikolaus Hansen [27]. These ES algorithms are great semi-automated self-adapting solutions for solving multi dimensional non-linear non-convex local optimization problems, even in rugged *search space* global optimization and multi-objective problems [28].

There is a strong tendency towards reducing the configuration complexity of heuristic algorithms. While some industry applications welcome the advantages, others do not

[3] a revised version of the paper is also available for download at the authors website with improved results obtained by correcting an implementation bug of the presented algorithm

entirely benefit from self-adaptation. In practical terms hybrid and reconfigurable implementations of different optimization algorithms can still hold a high value in the industry due to their capacity to easily reconfigure and adapt, following user experience, to specific traits of different black box problems. In many self-adaptation algorithms user reconfiguration can often be layered with added complexity, and long algorithm adaptation times unnaceptable. In response, algorithms with higher configuration complexity can be trained under hyper heuristics to enable more competitive performance, in this sense resembling embedded self-adaptation behavior. Some authors claim highly configurable heuristics transform the problem of parameterizing the function into a problem of parameterizing the heuristic. In practical terms there is a trade off between attaining the knowledge required to parameterize the heuristic system with acceptable results, and attaining the knowledge to understand a *black box* system.

A different approach methodology that could be applied to solve optimization problems is refered to Constraint Logic Programming (CLP) or simply Constraint Programming (CP) [29] [30]. CP is a programming paradigm where relations between variables are stated in the form of constraints on variables over a given domain. A solution for a Constraint Satisfaction Problem using CP is an assignment of a value to each of the variables that satisfies all the problem constraints. CP is typically used for solving decision problems but can also be extended to solve optimization problems by defining an evaluation function that maps the problem variables into a real value. The optimization process consists in a process that, after achieving a given valid solution, posts a new constraint stating that a new solution must have a higher/lower value (maximization/minimization problems) that the one previously achieved. It was not researched further as a viable solution for expeditious modeling parameterization due to personal decisions related to lack of experience with the methodology and time constraints. Its level of applicability in this specific, and somewhat neglected, field might be worth researching in future projects. However CP is typically more tailored for solving decision problems and not optimization problems like the ones analyzed on this paper.

The presented hybrid heuristic optimization algorithm combines traits of RCGA, SA and HS configurable to match the behavior of some of these methods and also double as a hyper heuristic to find its own best configuration for the problem to solve if required. Some academics might also find some traits of Particle Swarm Optimization [24].

A brief description of the algorithms combined in the hybrid approach follow.

2.1 Simulated Annealing

SA is inspired by the natural process of annealing of metallurgic materials. A slower cooling process would typically originate stronger ligaments than by rapid cooling process. This is explained at an atomic level by the fact that slower cooling allows larger elements to arrange them selfs in a more efficient configuration before the remaining elements connect around them. Generating a tighter configuration with stronger connections.

The algorithm was first presented by Kirkpatrick [17] originating from an adaptation of the Monte Carlo methodology applied to thermodynamic systems that would become referred as the Metropolis-Hastings algorithm [31]. Kirkpatrick, Gelatt and Vecchi [17] defined SA as a meta heuristic algorithm that would search new neighboring solutions of the combinatorial problem and accept them based in the formula from Equation 1.

$$p(\Delta f, T) = \begin{cases} e^{\frac{-\Delta f}{T}} & \Delta f \leq 0 \\ 1 & \Delta f > 0 \end{cases} \tag{1}$$

The stopping criteria for SA is usually passing a certain temperature threshold, given that the temperature value T decreases on each passing iteration by a certain rate which can be variable to better fit the specific problem. The probability of accepting a solution of lower quality decreases with the temperature. This process gives room for solutions to wander through a wider range of the search space during initial iterations and slowly reduce that wandering capacity, turning the algorithm into a standard hill climbing in the final iterations. This behavior helps avoiding a typical flaw where the optimization process gets stuck in a *local optimum* neighborhood.

The downside of SA is the inability to guarantee that a global optimum, instead of a local optimum, can always be reached or how many iteration steps it can take to guarantee it. These questions highly depend on the complexity of the problem and the linear dependency between the parameters and the quality function. The temperature descent ratio is also required to be fine tuned paying special attention to the characteristics of the problem. Uncareful parameter tuning can easily affect the performance of the algorithm, reducing it to the same of a basic hill climbing implementation.

Some authors have achieved higher performance by combining implementations of SA with the previously described tabu search (TS) algorithm. The work of Mishra [32] for instance is a good example of a successful application.

2.2 Real Based Genetic Algorithm

Genetic Algorithm is the name given to the nature inspired system based on the Darwinian concept of the survival of the fittest, where each solution to our combinatorial problem is uniquely encoded and part of a group or family of solutions which evolves by each passing generation. The individual solutions are crossbred and mutated to give birth to new generations of solution families sharing traits with the *originalparent* family of solutions. Original reference to this algorithm dates back to the work of Holland in 1975 [19]. A more modern reference can be found in the work of Goldberg [21].

Several implementations of the algorithm exist. Most implementation found in the literature follow the original concept of binary encoding and elitist selection of each new generation. The standard implementation refers to the problem's parameters being encoded into a binary dna string. Each possible parameter combination - our solution - is defined by a unique dna string which can be crossbred with other dnas or mutated alone to generate new dna solutions, yielding composite traits from the parents. There are quite many variants to the algorithm referring many possible ways to crossbreed the binary encoded solutions and the influence of the mutation factor in different types of problems.

A comparatively recent and quite interesting approach to the original genetic algorithms concept is the so called real-value (also referred to as real coded or real based) variation of the algorithm, handling continuous parameters, where the parameters no longer constitute a binary dna, instead they are stored in a vector, but the standard genetic algorithmic steps of parent selection, cross breeding, mutation and validation still take place.

Real based genetic algorithms have been taking a strong interest in the past decade, the hand book of genetic algorithms by Davis [33] contains a good general reference and some new developments to the work on this area made by Michalewicz [20]. Additionally, the work of Arumugama [34] is worth consulting for some information regarding new real based cross breeding techniques and their comparative work. Also worth mentioning are a couple of new real based cross over and mutation operands presented under the work of Deep and Thakur [35].

2.3 Harmony Search

Harmony Search (HS) is a fairly recent meta heuristic algorithm. It takes it's principle from musician improvisation sessions where musicians try different note combinations to find the best melody. It's theory and industry applications are fairly well documented in the literature by Geem [36], Lee [22] and Mahdavi [23].

HS is applicable to problems handling discrete and continuous variables alike, fitting a middle range of applicability where most meta heuristic algorithms can only handle discrete values and gradient-based mathematical algorithms can only be applied to continuous variables.

The algorithm works by initializing a random population of possible vector *melody* solutions, each solution is a combination of *notes* under a certain possible range. The algorithm uses a probability threshold to decide if part of the new solution vector is copied from previous memory of good performances or instead randomly improvised. Additionally, each *note* can have it's *pitch* slightly randomly adjusted, within a certain *pitch bandwidth*, depending on another probability threshold. Improved solutions are kept in memory and the process is repeated a certain number of iterations.

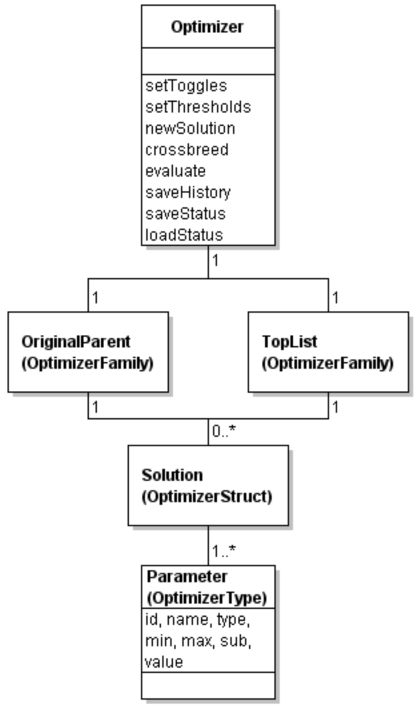
Some implementations of the HS algorithm vary the threshold probabilities or the *pitch bandwidth* during the iterative process to condition the algorithm into a simulated annealing alike behavior. Depending on the characteristics of problem, this variation can be very effective in providing improved results over the standard HS implementation.

2.4 Architecture Overview

The system was developed to solve generic *black box* expeditious modeling problems. The requirements include: handling the input of boolean, integer and real based parameters; being configurable by XML[4]; supporting hyper heuristic recursivity. As shown on Figure 1 the foundations of the optimizer rely on a population based system with basic principles of RCGA. There are two families resident in memory at all times, these are referred to as the *originalparent* family and the *toplist* family. The size of either of these families can be configured by XML to adapt the hybrid meta heuristic to the complexity of different problem types. Each new generation is obtained by cross-breeding the *originalparent* family with a chosen member of the *toplist* family.

[4] following markup language standards to represent our configuration data can enable higher interoperability with other expeditious modeling systems and invocation by script languages

Fig. 1. Optimizer architecture.



2.5 Thresholds

There are several thresholds present that incorporate Monte Carlo [31], HS and SA ideologies in the algorithm. We refer to the term *globalentropy* as a ratio (3) that will induce different levels of anarchy in the algorithm as it increases slightly every generation. Some of these thresholds determine the probability of having an entirely random (4) or partially random new solution (4).

$$globalentropy = iterationstep/maxsteps \tag{2}$$

$$rand() * globalentropy \ < \ trns \ : \ randSol() \tag{3}$$

$$rand() * globalentropy \ < \ trnt \ : \ randType() \tag{4}$$

Another kind of threshold referred to as toplist dispersion (6) defines the dispersion range of the *toplist* available for crossbreeding, this can be be comparable to the roulette wheel parent selection operator of RCGA. One other threshold referred to as value dispersion (6) defines the percentage of influence the *toplist* parent (*tlv*) has on the crossbreeding process with the *original* parent (*orv*). This process can be comparable to a hybrid version of the RCGA operands of average convex cross over and direction based cross over [33].

$$victim = (rand() * ttld * maxFamilySize) \tag{5}$$
$$newvalue = (ttvd * orv) + ((1 - ttvd) * tlv) \tag{6}$$

Other thresholds are present to determine the probability of influence of a directional vector from the *toplist* on the breeding process (13), similar to many mathematical gradient based implementations and recent optimizations to the acclaimed CMA-ES [27]. One other threshold (10) determines the jittering scope of the crossbreed value, similar to the *pitch* adjustment of HS and several RCGA mutation operands described in the literature.

$$scope = \|(orv - tlv)\| \tag{7}$$
$$igl = (1.0 - globalentropy) \tag{8}$$
$$ttvss = ttvss * ttvsv * igl \tag{9}$$
$$range = MaxParamValue - MinParamValue \tag{10}$$
$$scope < ttvss * range\ :\ maxscope = range * ttvsv \tag{11}$$
$$scope > ttvss * range\ :\ maxscope = scope * ttve \tag{12}$$
$$maxscope = scope * rand() \tag{13}$$
$$newvalue = newvalue + scope - (maxscope/2) \tag{14}$$

The threshold *ttvsv* (12) has a small effect on the jittering of the final solution which might only be worth fine tuning in very rare cases. For most of the tested development of the algorithm its value has been default to *globalentropy*.

Each of these thresholds can be configured (referred to as as *toggles*) to have a constant value or to follow a dynamically changing function. There are several of these different functions implemented into the system, each of them programmed to progress from a value of zero in the starting iteration to a value of one in the final iteration step. The most basic available function is the linear descent (*globalentropy*). Other available functions are adaptations of log and sin functions. Also available are the inverse functions of each of them.

The advantages of having a hybrid system with many *thresholds* to configure and multiple *toggles* to switch is the capacity of reusing the same system with many different type of algorithmic behavior configurations. The different configurations emulate different standard algorithm without having to embed complex libraries or several third party implementations of the usual algorithms to test which might work best for each type of problem.

A disadvantage of having many *thresholds* and *toggles* lies in the requirement to always reconfigure the system for good performance depending on the problem. Users with a lack of experience, internal knowledge or attention to detail can very well be replacing the original parameterization problem by another, the parameterization of the heuristic that can solve that problem faster. This raises questions that the heuristic system should be automated, following the research tendency of evolution strategies [26] towards self adaptation. Self automated systems are harder to implement and predict results, the extra function evaluation time required for the adaptation to reach the optimum is also often an unacceptable trade off. Expert configuration of a non adaptive heuristic system can still often deliver faster results.

2.6 Performance Tests

It is imperative to note that despite the effort directed to ensure competitive performance under standard non-convex non-linear problems of continuous nature, the hybrid system presented in this paper was not at all aimed to outperform any of the state of the art optimization algorithms. The system was developed as a lightweight, portable, reconfigurable and flexible prototype to embed in expeditious modeling systems. As previously stated, expeditious modeling systems have additional requirements beyond performing competitively against pre-defined sets of non-convex non-linear problems of continuous nature. Expeditious modeling optimizer systems are required to equally assist competitively in solving problems of different type and complexity, with mixed discrete and continuous parameters. Overall, the system was aimed to provide a solution capable of performing both the bluntly configured short iteration phases aimed to determine generic guidelines to solve an optimization problem, and the extremely fine tuned configuration for long run convergence for obtaining the global optimum of complex problems.

Regardless of these requirements, the performance of the system was tested against some of the standard batch of non-convex non-linear continuous problems from the IEEE Congress on Evolutionary Computation 2005 [37]. The algorithm was over fitted for one of the standard problems, the *Rosenbrock*, at $D = 10$ dimensions[5] and then tested on the *Sphere*, *Schwefel 1.2*, *Rosenbrock*, *Rastrigin* and *Weierstrass* at $D = 10$ and $D = 30$.

The average results for 25 runs of each problem can be consulted in Table 1. The first column shows the name of the problem tested while the following the average from 25 runs, at steps 2% 40%, 80% and 100% of the optimization process, the final column the standard deviation of the final result gives a good indicator at how spread the 25 solutions were from the average value. The first 5 problems were executed at $D = 10$, with a generation of 10, executing $10*10^5$ function evaluations, the last 5 were executed at $D = 30$, with a generation of 30, executing $30 * 10^5$ function evaluations. The original CEC'05 Special Session on Real-Parameter Optimization [37] consisted of benchmarking 11 algorithms each optimizing 25 different test functions under $D = 10$ and $D = 30$. Measuring which ones would solve the given problems under the target precision error of 10^{-8} under each function for $D * 10^5$ function evaluations.

The presented table does not relate directly to the results obtained from CEC'05 session [27] but analysing those results and comparing them to the performance of our algorithm allows us to conclude that despite our hybrid performing quite below the state of the art, it has an acceptable performance optimizing non-convex non-linear types of problems.

[5] all global optimization problems from the batch have been specifically developed as a calculus sum of finite dimensions. increasing the number of dimensions exponentially increases the complexity of the problem

Table 1. Average quality function at different stages of the convergence.

problem	2% μ	40% μ	80% μ	final μ	final σ
Sphere	2.880249e+004	5.069603e-007	7.443773e-010	2.746327e-016	1.389618e-016
Schwefel 1.2	4.834547e+005	3.512164e-006	6.751018e-009	1.194683e-015	4.648368e-016
Rosenbrock	3.028539e+010	1.190515e+003	3.619151e-002	3.720837e-005	1.161202e-004
Rastrigin	1.533455e+002	3.236556e-005	9.360353e-008	3.327253e-014	1.758198e-014
Weierstrass	1.463048e+001	2.082120e-001	4.062040e-002	2.013143e-002	9.980064e-002
Sphere	1.133168e+005	7.510152e-005	7.610557e-008	5.908891e-012	3.502260e-012
Schwefel 1.2	2.520423e+007	1.522236e-002	1.362870e-005	6.535891e-010	2.939757e-010
Rosenbrock	1.412603e+011	4.283153e+002	9.933642e+000	3.956425e+000	1.151809e+001
Rastrigin	5.652055e+002	7.956591e-001	8.582548e-006	6.416542e-010	3.106761e-010
Weierstrass	4.915724e+001	1.468030e+000	4.400829e-001	1.226722e-001	1.910961e-001

3 EXPEDITIOUS MODELING

An example application of an expeditious modeling problem which the hybrid heuristic system can solve is the estimation of corrupt or unavailable data. A specific test case was prepared using the XL3D modeler [6] [38]. The test case involved the parameterization of production rules to determine missing height values from a set of buildings. Expeditious modeling systems typically require a geographically referenced database of the model and a set of rules. These modeling rules generate the geometry of the model according to the referenced data.

In our specific test case the known information for each building included the values of the perimeter, area and its *bottomzvalue* [7]. To model a building we also require to know its height, which is missing data from our database. An extra modeling rule estimates the height of a building, conceptually implying a possible relation between the height and the building's perimeter (15), area (16), and *bottomzvalue* (17). An additional variable *avgz* (18) refers to the average z value of each building from the set. The applied formula is described mathematically in Equation (19).

$$ap = (crp - 1) * fcp * perimeter \tag{15}$$
$$av = (cra - 1) * fca * area \tag{16}$$
$$ab = (crb - 1) * fcb * bottomzvalue \tag{17}$$
$$avgz = bottomzvalue + averageheight \tag{18}$$
$$topzvalue = avgz + disp * (ap + av + ab) \tag{19}$$

The formula has a total of 7 unknown fields to be parametrized:

- $disp \in [0.0..1.0]$, the dispersion rate from the average height;
- $crp \in [0..3]$, perimeter correlation signal;
- $cra \in [0..3]$, area correlation signal;
- $crb \in [0..3]$, *bottomzvalue* correlation signal;
- $frp \in [0.0..1.0]$, perimeter correlation factor;

[6] an expeditious modeling system based on L-Systems applied to modeling urban environments
[7] the height of the base of the building

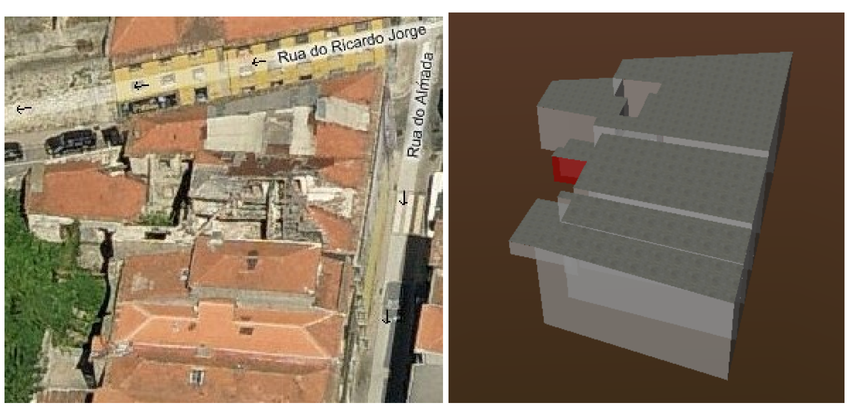
- $fra \in [0.0..1.0]$, area correlation factor;
- $frb \in [0.0..1.0]$, *bottomzvalue* correlation factor.

It is important to note that this is a simple empiric attempt to determine a sense of a possible correlation. Clearly the features of all buildings would not ever follow this exact relational formula, and in this sense, more accurate formulations could be easily written by professionals with architectural background and local domain knowledge, to obtain better results. This formula is merely a demonstration that even basic tests with crude relational formulas can greatly assist in the estimation of realistic values for expeditious modeling, allowing the application of a more sensible methodology over common trial and error methods.

In this specific test a very small data set area was collected from half a dozen typical buildings in a corner of *Rua do Almada* in the city of Porto. This data was used to determine our *avgz* and calibrate the production rules. The function that measured the quality of the modeling rule compared the height differences of the buildings with known values and their procedural formula estimation counterpart. A minimization process was applied in two phases of short iteration steps. The first phase with a large *originalparent* family size to find the best correlation signals for the system, the second with a shorter family size to find the global optimum under those correlation signal constraints.

On the left side of Figure 2 we can see a satellite overview image of the actual area obtained from Microsoft Visual Earth[8], on the right side the rendered model screenshot. The buildings in grey are rendered with real height values, while the buildings in red were rendered by applying the calibrated modeling rule estimating an height value. After having calibrated the parameters of the height estimation rule, it may be applied

Fig. 2. Comparative images of Rua do Almada calibration area.



to other sections of the city with similar housing characteristics. This process requires

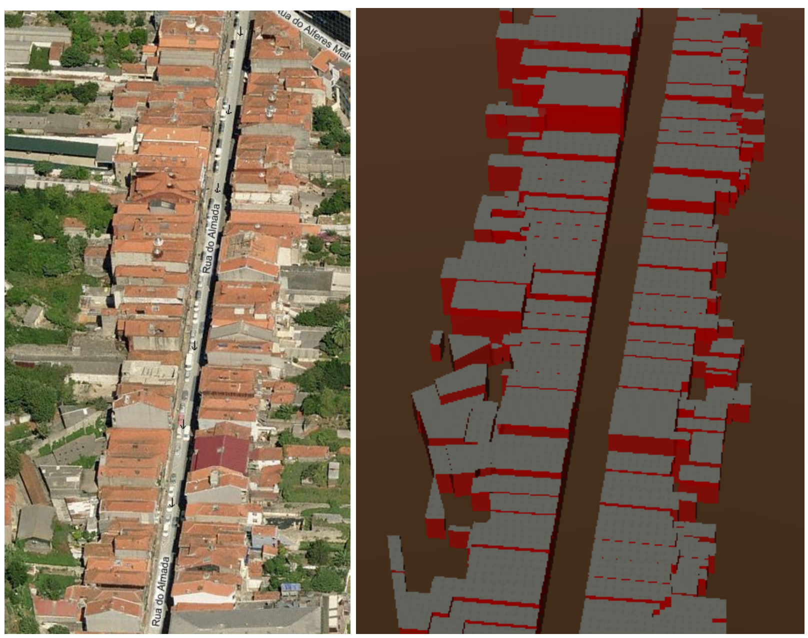
[8] http://maps.live.com

some attention to problems of over-fitting the calibration zone. In Figure 3 we have a different section of *Rua do Almada*, with buildings that resembles the characteristics of the small calibration area. On the left we have the Microsoft Visual Earth overview, on the right the generated model, all buildings generated using the previously calibrated height estimation rule:

$$avgz + 0.15623 * (0.26615 * area)$$

Comparing the images and analysing the models, it is obvious that the estimated height

Fig. 3. Comparative images of Rua do Almada generated area.



from some of the buildings does not match the real height. However, a large percentage of the buildings does present an acceptable realism. We can conclude that the process is far from being perfect. Better selection criteria for estimated modeling rules and attention to standard over-fitting calibration problems could easily be added to significantly improve these results.

The proposed methodology does not completely solve the problem but does offer a good ground work alternative, considerably useful for problems where accurate data or expert modeling assistance is unavailable to generate a realistic environment in a short period of development time.

4 CONCLUSIONS AND FUTURE WORK

Different methods can be applied to determine adequate production rules for expeditious modeling systems, ranging from simple empirical definitions to observing 3D modelers creating objects. Fine tuning the parameters of these production rules is a big problem requiring domain knowledge, attempting to fine tune them by trial and error is unwanted.

An hybrid heuristic optimization algorithm was developed to assist in automatic parameterization for expeditious modeling of virtual urban environments. The system is a hybrid version of RCGA with significant traits of SA and HS and was developed to solve black box type of global optimization problems that require boolean, integer and float type of input parameters. The algorithm was not developed to out perform the state of the art non-convex non-linear optimization algorithms but tests with the standard batch of problems from CEC'05 proved an acceptable performance under the problems of continuous nature.

The system was applied to a real test case scenario in an expeditious modeling system. The test case aimed to determine the optimum parameter combination of certain productions rules through comparing modeling results with values from real data. The calibrated production rules were then applied to other areas of similar characteristics and successfully generated realistic urban models. This test case despite its relatively low complexity, successfully demonstrated some of the potential of the developed system.

There are many applications of the proposed system, its reconfigurability and architecture were developed with special attention to the multiplicity of global optimization problems that appear in expeditious modeling. As such, it would be interesting to observe the performance of the system operating under more complex test cases than the one documented in this paper.

Clustering problem types and applying hyper heuristics to define the optimum configurations of the hybrid heuristic for each problem class could be a useful addition to the system.

Other application ideas are to expand the system to an interactive feedback methodology where the system would not determine the quality fitness of each solution automatically but instead allow interactive selection, common to many generative computation applications. This would complicate the fine tune process of certain parameters with quantifiable solution quality but at the same time allow a certain creative freedom with more applications. The more obvious one would be assisting to define constraints for parameters with undeterminable quality fitness functions, this could be due to a lack of knowledge on the production rules or an incapacity to measure quality improvements mathematically. Another application would be in exploring the capacities of the production rules, there can be scenarios in expeditious modeling where the user could warrant something different but do not quite know exactly what should be different until they can see it taking form. In these cases, interactive selection as a way to determine the quality of the solutions, would be a valuable addition.

References

1. Schilling, A., Coors, V.: 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. (2003) 39–48
2. Ito, H., Teh, S., Nakanishi, H., Tagawa, T.: Design and implementation of 3d interface for digital city. In: Electronics and Communications in Japan (Part III: Fundamental Electronic Science), number 6. Volume 88. (2005) 60–68
3. Hildebrand, A., Dahne, P., Seibert, F., Christou, I., Demiris, A., Diorinos, M., Ioannidis, N., Almeida, L., Diogo, A., Weidenhausen, J.: Augmented reality based system for personalized tours in cultural heritage sites. In: International Conference of Augmented Reality, Virtual Environments and 3D Imaging Proceedings. (2000)
4. Zach, C., Klaus, A., Bauer, J., Konrad, K., Grabner, M.: Modeling and visualizing the cultural heritage data set of graz. In: Proceedings of the 2001 Conference on Virtual Reality, Archeology, and Culture Heritage. (2001) 219–226
5. Wonka, P., Wimmer, M., Sillion, F., Ribarsky, W.: Instant architecture. In: Proceedings of ACM SIGGRAPH 2003 / ACM Transactions on Graphics. Volume 22., ACM Press (July 2003) 669–677
6. Shlyakhter, I., Rozenoer, M., Dorsey, J., Teller, S.: Reconstructing 3d tree models from instrumented photographs. In: IEEE Computer Graphics and Applications, number 3. Volume 21. (2001) 53–61
7. Faugeras, O., Laveau, S., Robert, L., Csurka, G., Zeller, C.: 3-d reconstruction of urban scenes from sequences of images. In: Rapport de Recherche 2572, INRIA (1995)
8. Pollefeys, M., Proesmans, M., Koch, R., Vergauwen, M., Gool, L.: Acquisition of detailed models for virtual reality. In: Virtual Reality in Archaeology. Volume 843., BAR International Series, CAA, Archaeopress, publishers of British Archaeological Reports (1998) 71–77
9. Muller, P., Wonka, P., Haegler, S., Ulmer, A., Gool, L.: Procedural modeling of buildings. In: Proceedings of ACM SIGGRAPH 2006 / ACM Transactions on Graphics. Volume 25., ACM Press (2006) 614–623
10. Lindenmayer, A.: Mathematical models for cellular interaction in development, parts i and ii. In: Journal of Theoretical Biology. Volume 18. (1968) 280–315
11. Prusinkiewicz, P.: Graphical applications of l-systems. In: Proceedings of ACM SIGGRAPH 1986, ACM Press (1986) 247–253
12. Parish, Y., Muller, P.: Procedural modeling of cities. In: Proceedings of ACM SIGGRAPH 2001, ACM Press (2001) 301–308
13. Cruz, F., Coelho, A., Reis, L.: Automatic parameterization for expeditious modelling of virtual urban environments: A new hybrid meta heuristic. In: Proceedings ICINCO 2008. (2008)
14. Russel, S., Norvig, P.: Artificial Intelligence: A modern approach. 2nd edn. Prentice-Hall (2003)
15. Jacobson, S., Yucesan, E.: Analyzing the performance of generalized hill climbing algorithms. In: Journal of Heuristics, number 4. Volume 10., Hingham, MA, USA, Kluwer Academic Publishers (2004) 387–405
16. Gendreau, M.: An introduction to tabu search. In: Handbook of Metaheuristics. Series: International Series in Operations Research and Management Science, Chapter 2, Kluwer Academic Publishers (2003) 37–54
17. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimizing by simulated annealing. In: Science 4598. Volume 220. (1983)

18. Aarts, E., Korst, J., Michiels, W.: Simulated annealing. In: Search Methodologies, Springer (2005) 187–210
19. Holland, J.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. MIT Press (1992)
20. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag (1994)
21. Goldberg, D.: The Design of Innovation: Lessons from and for Competent Genetic Algorithms. Addison-Wesley, Reading, MA (2002)
22. Lee, K., Geem, Z.: A new meta-heuristic algorithm for continuous engineering optimization harmony search theory and practice. In: Computational Methods Applied Mechanical Engineering. Volume 194. (2005) 3902–3933
23. Mahdavi, M., Fesanghary, M., Damangir, E.: An improved harmony search algorithm for solving optimization problems. In: Applied Mathematics and Computation 188. (2007) 1567–1579
24. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Neural Networks, 1995. Proceedings., IEEE International Conference on. Volume 4. (1995) 1942–1948
25. Dorigo, M.: Optimization, Learning and Natural Algorithms. PhD thesis, Politecnico di Milano, Italy (1992)
26. Eiben, A., Schoenauer, M.: Evolutionary computing. In: Information Processing Letters. Volume 82. (2002) 1–6
27. Hansen, N.: The cma evolution strategy: a comparing review. In: Towards a new evolutionary computation. Advances on estimation of distribution algorithms, Springer (2006) 75–102
28. Igel, C., Hansen, N., Roth, S.: Covariance matrix adaptation for multi-objective optimization. In: Evolutionary Computation, number 1. Volume 15., MIT Press (2007) 1–28
29. Dechter, R.: Constraint Processing. Morgan Kaufmann Publishers Inc. (2003)
30. Apt, K.: Principles of Constraint Programming. Cambridge University Press (2003)
31. Metropolis, N., Ulam, S.: The monte carlo method. In: Journal of the American Statistical Association, number 247. Volume 44. (1949) 335–341
32. Mishra, N., Prakash, M., Tiwari, R., Shankar, F., Chan, T.: Hybrid tabusimulated annealing based approach to solve multi-constraint product mix decision problem. In: Expert Systems with Applications 29. (2005)
33. Davis, L.: Handbook of Genetic Algorithms. Van Nostrand Reinhold (2001)
34. Arumugama, M., Raoa, M., Palaniappanb, R.: New hybrid genetic operators for real coded genetic algorithm to compute optimal control of a class of hybrid systems. In: Applied Software Computing 6. (2005) 38–52
35. Deep, K., Thakur, M.: A new mutation operator for real coded genetic algorithms. In: Applied Mathematics and Computation. Volume 193. (October 2007) 211–230
36. Geem, Z., Kim, J., Loganathan, G.: A new heuristic optimization algorithm: Harmony search. In: Simulation 2. Volume 76. (2001) 60–68
37. Suganthan, P., Hansen, N., Liang, J., Deb, K., Chen, Y., Auger, A., Tiwari, S.: Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. Technical report, Nanyang Technological University, Singapure (2005)
38. Coelho, A., Bessa, M., Sousa, A., Ferreira, F.: Expeditious modeling of virtual urban environments with geospatial l-systems. In: Computer Graphics Forum, number 4. Volume 26. (2007) 769–782