An Agent-based Framework for Intelligent Optimization of Interactive Visualizations

Pedro Miguel Moreira[1,2], Luís Paulo Reis[2,3] and A. Augusto Sousa[4]
[1]*ESTG-IPVC, School of Technology & Management, Polytechnic Institute of Viana do Castelo, Viana do Castelo, Portugal*
[2]*LIACC-Artificial Intelligence and Computer Science Lab., Univ. of Porto, Porto, Portugal*
[3]*EEUM-School of Engineering, Univ. of Minho-DSI, Campus de Azurém, 4800-058 Guimarães, Portugal*
[4]*INESC-TEC / DEI - Faculty of Engineering, Univ. of Porto, R. Dr. Roberto Frias, 4200-465 Porto, Portugal*

Keywords: Optimization, Multi-Agent Systems, Visualization.

Abstract: Interactive visualization of virtual environments is an active research topic. There is a multiplicity of applications such as simulation systems, augmented and mixed reality environments, computer games, amongst others, which endlessly demand for greater levels of realism and interaction. At every stage of the process, including modeling, image synthesis, transmission and navigation, there are identifiable circumstances which may compromise the achievement of high quality solutions for the posed problems. For many of these problems, an effective use of optimization tools can play a major role in order to achieve solutions with better quality. Within this context, an innovative optimization architecture is presented regarding to two major principles. The first principle comprises the possibility to integrate, with reduced effort, the optimization tools with existent applications and systems. Thus, we propose an agent-based framework where the optimization application may operate as an independent process in respect to the visualization application where communication is achieved by means of a specifically developed high-level message based protocol. The second principle establishes on the utilization of a class of intelligent optimization methods, known as metaheuristics, which major distinguishing quality is their great level of problem-independence, thus, enabling a wider application. The paper describes conducted experiments and presents results that demonstrate the utility and efficacy of the proposed framework.

1 INTRODUCTION

There are many visualization related tasks in the context of virtual environments that pose problems which may benefit from the adoption of optimization tools. Although, the works found in the literature typically tend to embed optimization deeply in the application. Such course of action conducts, as a result, to design and development efforts that are likely to have a limited reuse, besides the context for which they were taken.

In this paper we introduce an agent based optimization framework aiming to provide visualization related problems the ability to interact with an optimization agent in order to obtain optimized solutions. The target applications are those that include some kind of hard problems, e.g. combinatorial optimization, for which there is not an efficient algorithm to achieve optimal solutions.

A key requirement of such agent-based framework is to provide tools to attain, as much as possible, a seamless integration to existing visualization applications or related algorithms. Thus, a minimal redesign or re-code effort is expected in order to make use and take benefit of such tools. Besides, the agent based architecture innately qualifies for remote operation with respect to the optimizer and the applications. Communication and interoperability is achieved via a high-level message protocol over a standard (or existent) communication infrastructure (e.g. a computer network with TCP/IP). The remote operation capabilities open a wide range of application scenarios such as the deployment on heterogeneous target platforms (e.g mobile) or a better load balancing between optimization and visualization tasks using a distributed architecture, as depicted by Figure 1.

The herein presented agent-based framework endeavors applicability and utility for a wide range of distinct problems with reduced effort and significant levels of reuse. As a result, the strong decoupling

Moreira P., Reis L. and Sousa A..
An Agent-based Framework for Intelligent Optimization of Interactive Visualizations.
DOI: 10.5220/0004185301540159
In *Proceedings of the 5th International Conference on Agents and Artificial Intelligence* (ICAART-2013), pages 154-159
ISBN: 978-989-8565-38-9
Copyright © 2013 SCITEPRESS (Science and Technology Publications, Lda.)

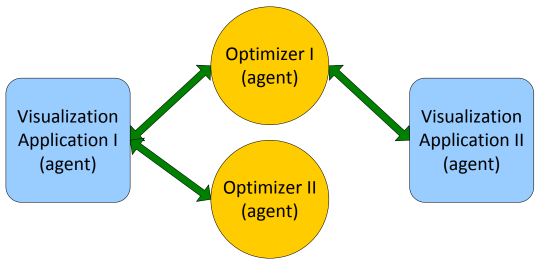

Figure 1: Agent-Based Optimization Architecture.

between the optimization process and the problem comes into a major requirement. Ultimately, the framework endeavors the application with a minimal amount of domain specific knowledge. To fulfill such requirements it is established on general purpose intelligent optimization methods, specifically metaheuristics, that are made available through optimization agents. Metaheuristics have several interesting features(Blum and Roli, 2003), that can be summarized as folllows: they guide the search process making use of techniques ranging from basic local search to complex learning processes; they are defined at a high level of abstraction, thus they are not problem-specific; they can usually find an optimal or near-optimal solution in bounded time; they may use forms of memory, thus benefiting from acquired search experience; they make use of randomized processes in order to promote diversification and to increase robustness; and they are often inspired by natural phenomena (biological, physical, ethological, etc).

2 FRAMEWORK DESIGN

This section introduces the agent-based framework design methodology, stating its main principles, goals and requirements. The main requirements of the proposed optimization framework are based on the following observations:

- There are many problems related to visualization and rendering tasks that can benefit from the use of optimization. Some of these problems are difficult in the sense that there is not a known algorithm that can obtain optimal solutions efficiently.
- In the visualization and computer graphics domains, researchers often develop their own optimization systems. This happens mainly because, as there is not any generic framework, it turns easier to develop a good optimization system from scratch. However, having such a generic optimization system shall enable the developer to test using different optimization methodologies for each visualization problem.
- When used, optimization techniques are tightly and deeply entangled in the visualization or rendering algorithms resulting in an increased intricacy of them. As this course of action does not promote code and design reuse, bigger efforts are required to test different optimization techniques and to apply similar optimization techniques to new problems.
- Graphical applications are regularly computationally intensive. Moreover, optimization applications are also computationally intensive. Therefore, the separation of the visualization from the optimization tasks may be a way to improve the overall performance. These tasks may be executed at different machines (or cores) enabling the the workload to be distributed and increasing the overall performance thus.

As a result, and in order to overcome the aforementioned problems and limitations, the proposed methodology has the following fundamental principles, which will be further developed in the next sections respecting to the design and implementation.

The optimization process must:

- Be as independent as possible from the application domain to permit its applicability to a wide range of problems;
- Provide a conceptual decoupling from the visualization application;
- Be implemented as autonomous agents; provide with suitable interoperability and communication mechanisms;
- Provide a set of algorithms and tools to make possible to test and compare the application of different algorithms and parameters;
- Permit the easy integration and reuse of new optimization algorithms and modules.

3 FRAMEWORK IMPLEMENTATION

The framework core architecture is composed of six core sets of classes, in respect to their responsibility on the optimization process. The proposed and implemented architecture is depicted in Figure 2. A description of each set of classes follows.

MetaHeuristics Classes. The MetaHeuristics set of classes is defined the different metaheuristics algorithmic templates. These classes are explicitly designed to do not have a particular representation of the problem. They delegate on other classes to perform generic actions, such as neighborhood

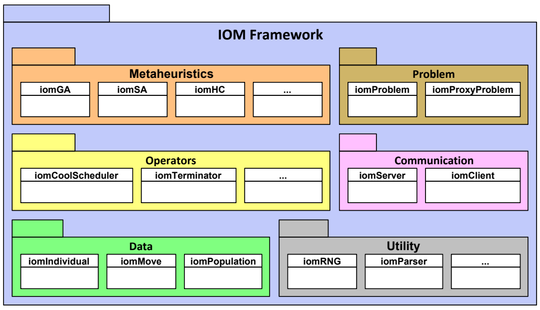

Figure 2: Core sets of classes.

selection, mutation, cross over, or termination criteria. Implemented and available classes include algorithmic templates for single point metaheuristics, concretely Hill Climbing and Simulated Annealing(Kirkpatrick et al., 1983; Cern, 1985), and also for a population based metaheuristic - Genetic Algorithms(Holland, 1975). Variations to a given metaheuristic can be achieved by instantiating them with operators performing distinct behaviors or strategies. New metaheuristic algorithmic templates can be easily integrated into the framework by inheriting from the abstract base Metaheuristic class.

Operator Classes. Operators are mid level classes which implement a vast set of components and actions required by metaheuristics. Examples are neighborhood explorers, termination criteria and selection mechanisms. It is provided a set of abstract template classes for each of these operators together with a set of concrete classes implementing commonly used operators.

Data Classes. This set of classes is designed to hold of the basic data structures such as fitness representation, individuals (solutions are treated as individuals with an associated fitness value) and moves. They have to be implemented in dependence to the problem. Usual data representations, such as vectors, are provided. Data classes are not intended to be used as algorithmic components and therefore they do not have any specific methods or functionality than those required related to data handling and basic (problem dependent) operation.

Problem Classes. Problem classes are deliberately designed apart from the Metaheuristics to enable an easy integration with an external application or agent. They can be thought as the communication gateway between the optimizer and application agents. They have three fundamental responsibilities: hold a state of the problem; handle the messages interchanged between the optimizer and application; and evaluate solutions.

Communication Classes. These classes provide the system with communication capabilities. The implemented classes make use of TCP/IP sockets and implement a custom communication protocol.

Utility Classes. In addition to the main core classes, the framework also provides the user with a set of utility classes such as random number generation, statistics, parsing, etc.

As mentioned before, a relevant and innovative feature of this optimization framework is concerned to the separated and autonomous execution of optimizer and the application. Optimizer and application are agents that can be executed on different machines given that they are interconnected by some network infrastructure.

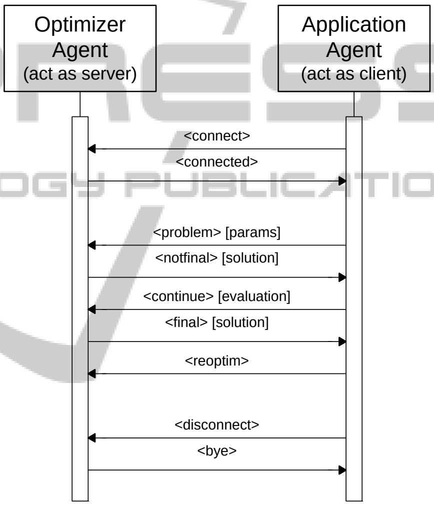

Figure 3: Communication Protocol.

A simple message protocol was designed in order to support inter operation between the two classes of agents (Figure 3). The protocol was intended to be architecture independent, so it does not limit its applicability. The application agent acts as a client for optimization purposes and initiates the connection by sending a `<connect>` to the optimization agent, that acts as a server. The optimizer agent responds with a `<connected>` message, acknowledging the connection. The application agent then requests optimization by means of a `<problem>` message encompassing parameters a problem description.

4 Experiments: Optimization of Multiple View Selection

RoboCup was created as an international research and education initiative, aiming to foster artificial intel-

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

ligence and robotics research, by providing a standard problem, where a wide range of technologies can be examined and integrated (Kitano et al., 1997). RoboCup Rescue project focus on urban disaster rescue. It aims at promoting research and development by involving multi-agent team work coordination.

RoboCup Rescue competitions can be difficult to monitor and visualize as they comprise a complex and dynamic environment (Certo et al., 2006).

Multi view visualizers intend to enable users and team developers with a more effective understanding of the on going action, since it is expected to automatically provide a set of distinct views, selected in order to optimize their relevance with respect to the understanding of the evolving simulation.

Selecting multiple views in virtual environments is interesting to many other applications. As, for instance, virtual museums or tourism in order to generate a best set of views over a set of relevant objects, or to generate paths for virtual exploration. Another relevant example is virtual cinematography, in order to automatically position, select and move virtual cameras.

The best set of views can be formulated as an optimization problem and it was chosen to conduct a first set of experiments in order to test and validate the previously described agent-based optimization framework.

4.1 Problem Formulation

In the context of RoboCup Rescue, the best multiple view problem can be informally stated as follows (Moreira et al., 2006). In an urban rescue environment there are m visualization agents that can obtain views over the scene. The problem is to find the set of k views that better describes the simulation at each moment. The visualization agents are controllable in the sense that one can affect their viewing parameters. In order to formulate this problem as an optimization problem, we developed a simple model for estimating the quality of a multi-view. The devised multiview quality is a function of the *visibility*, *relevance*, *redundancy* and *eccentricity* of the entities represented in the set of selected views. Thus, the corresponding optimization problem can be formalized as follows:

$$\text{MAXIMIZE:}Q(MV) =$$

$$= \sum_{j=1}^{k}\sum_{i=1}^{n} Vis(e_i^j)\cdot Red(e_i|MV)\cdot(W_1.Rel(e_i) - W_2.Ecc(e_i^j))$$

$$E = \{e_1, \dots e_n\}; V = \{v_1, \dots v_m\}$$

$$v_i = f(\vec{Pos}_i, \vec{VD}_i, V\vec{U}P_i, FoV_i)\ i \in \{1, \dots, m\}$$

$$MV = \{sv_1, \dots, sv_k\} \text{ where } MV \subset V \text{ and } sv_i \neq sv_j\ \forall i \neq j$$

In the given formulation, E denotes the set of n entities that have relevance in the scene (buildings, agents, etc) and V is the set of different views (equals the number of agents/entities with viewing capabilities). Each view is characterized by usual camera parameters, as position $\vec{Pos}_i$, view direction $\vec{VD}_i$, relative camera orientation $V\vec{U}P_i$ and field of view FoV_i.

A multi-view, MV, comprises a set of k distinct selected views (sv) from V.

The problem is to find the optimal MV set, with appropriate view parameters, that better describes the rescue scenario given a quality metric. We developed a quality Q metric using the following criteria (note that e_i^j denotes the visual properties of an entity e_i in a image obtained by the view j). The parameters $W1$ and $W2$ denotes constants used to adjust the relative weight of relevance and eccentricity.

Visibility: $Vis(e_j^i)$. This feature relates to the visibility of the relevant entities (e.g. the visible area). Several factors contribute to an entity's visibility such as the distance to the viewpoint, size, relative orientation, and also by how much partial occlusion it suffers from other objects.

Relevance: $Rel(e_i)$**.** A measure of how relevant is the entity for the purpose of the visualization. For example, if tracking emergency situations, a building on fire has a greater relevance than an unaffected building. The intrinsic importance of an object is also considered, e.g. hospitals, fireman headquarters, schools have more relevance than ordinary buildings.

Redundancy: $Red(e_i|MV)$**.** It is expected that the multiple set of views describe as much as possible distinct situations occurring during the simulation. Thus, redundant views over the same entities are penalized.

Eccentricity: $Ecc(e_j^i)$**.** A measure on how distant to the center of the image an object is displayed. This criterion has a perceptual foundation based on the observation that an user will focus attention to image centered entities rather than to those in more peripherical regions.

An expected multi-view is shown in Figure 4. As it is depicted, regions with very intense ongoing rescue operations are shown with greater detail. Simultaneously other ares covered by wider views configured in order to get a more comprehensive picture of (an area) of the rescue scenario.

Using larger sets of views may be also of interest, for instance, for surveillance centers, allowing to have a better coverage of the ongoing operations either in coverage either in detail. The above general problem formulation was used in the experimental setup with some simplifications. Namely, it was made $W2 = 0$,

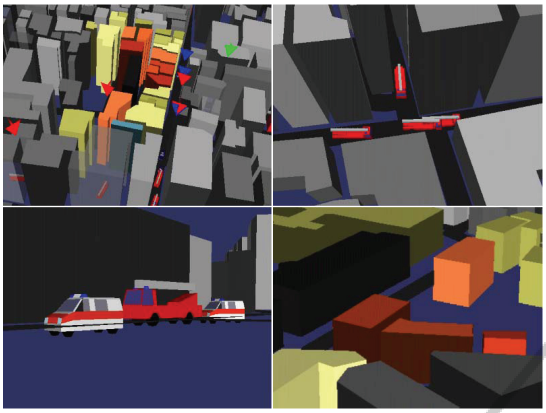

Figure 4: A multiview of a typical rescue scenario.

i.e. eccentricity was not considered, and agents were provided with omnidirectional viewing abilities as it is typical for RoboCup Rescue robots. Visibility was evaluated as approximately proportional to the inverse distance from the viewing entity to the viewed object, accounting for maximimum visibility of all entities near than 50 meters and null visibility for objects far than 500 meters. This model, although not complete, seems adequate with the viewing parameters used in RoboCup Rescue operations.

4.2 Experimental Setup

In order to test the applicability of the optimization methodoly and supporting architecture in respect to visualization problems, preliminary tests were conducted under the RoboCup Rescue domain in order to optimize a set of views over a rescue simulation scenario. For the purpose, simple RoboCup Rescue visualization problems were used. These problems were set up using distinct numbers of rescue entities (with distinct abilities) and different numbers of agents with viewing capabilities. Three different scenarios were configured with respectively 500, 200 and 100 agents with viewing abilities. For each of these scenarios three different problem instances were tested in order to find the best set of 4, 10 and 20 views. Each problem instance was solved by three different meta-heuristics: Hill Climbing (HC); Simulated Annealing (SA); and Genetic Algorithms (GA).

Each of the problems was solved using an evaluation function ignoring eccentricity as was told before.

The evaluation function (Equation 1) uses visibility, relevance and redundancy. Redundancy is used here in order to consider only the best view of each entity as its global contribution to the evaluation function. Accounting for redundancy, has the purpose to have a better coverage of the rescue arena, namely by avoiding a selection of a set of similar views (corresponding to spatially close agents) focused on the same entities.

Table 1: Experimental results using the second evaluation function.

Number of views	Algorithm	Number of entities: 500	200	100	
20	HC	3099.96	1290.33	678.62	MEAN
		3102.03	1291.53	679.55	MAX
		2.03	0.69	0.85	STDEV
	SA	3099.86	1289.34	679.55	MEAN
		3102.97	1291.93	679.55	MAX
		3.82	3.18	0.00	STDEV
	GA	3101.10	1291.69	679.55	MEAN
		3103.13	1291.93	679.55	MAX
		3.99	0.22	0.00	STDEV
10	HC	2646.48	1170.16	593.94	MEAN
		2654.90	1170.60	594.25	MAX
		10.17	0.60	0.36	STDEV
	SA	2650.63	1170.16	594.12	MEAN
		2654.90	1170.60	594.25	MAX
		3.24	0.60	0.28	STDEV
	GA	2653.75	1170.60	594.25	MEAN
		2654.90	1170.60	594.25	MAX
		2.58	0.00	0.00	STDEV
4	HC	2206.53	892.94	442.80	MEAN
		2207.21	892.94	442.80	MAX
		0.62	0.00	0.00	STDEV
	SA	2206.75	892.94	442.80	MEAN
		2207.21	892.94	442.80	MAX
		0.62	0.00	0.00	STDEV
	GA	2207.21	892.94	442.80	MEAN
		2207.21	892.94	442.80	MAX
		0.00	0.00	0.00	STDEV

$$\text{MAXIMIZE:}Q(MV) = \sum_{j=1}^{k}\sum_{i=1}^{n} Vis(e_i^j)\cdot Red(e_i|MV)\cdot W_1\cdot Rel(e_i) \tag{1}$$

A set of initial experiments was performed in order to tune the algorithms inherent parameters. For the Hill-Climbing algorithm, a random neighbor approach was used with the neighborhood being generated by changing one of the viewing agents by another. For Simulated Annealing, the same neighborhood was used and an exponential cooling schedule with an exponential ratio of 0.99 was used. For Genetic Algorithms, an elitist selection mechanism (keeping the best half of the population), single point random crossover, and 10% of mutation probability were used.

4.3 Experimental Results

A synthesis of the obtained results with the aforementioned evaluation function (Equation (1)) is presented in Table 1 according to the described experimental setup. The table displays the maximum (max), mean and standard deviation (sd) values of five complete experiments for each problem instance and algorithm. Results analysis demonstrate that genetic algorithms tend to achieve better global results, even using a similar CPU time (GA parameters were configured in order to use approximately the same CPU time as SA).

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Simulated annealing is also able of achieving better results if compared to the used hill-climbing.

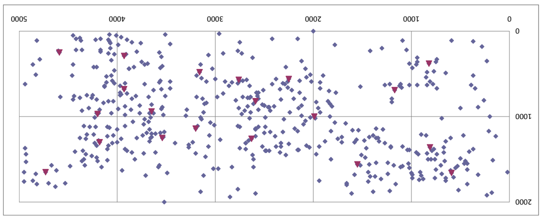

Figure 5: Graphical depiction of the achieved solution using the second formulation for a problem with 500_20, where Δ represent the selected viewing agents.

A graphical depiction of the solution is shown in Figure 5. It can be concluded that the selected viewing agents are now spread around and able to cover almost all the rescue arena. The set of solutions tend to cover the most relevant regions without (or, at least, less) repetition of the focused areas. Achieving good solutions for this problem is difficult mostly because the multi view quality is a sum of interdependent parcels. Results also demonstrated that the framework is able of optimizing best multiple view problems and instances with different dimensions with respect to the number of rescue entities and viewing entities. Another conclusion is that the choice for simpler metaheuristics, such as hill climbing, can achieve good results. However, slightly different formulations may lead to poor performance of this simpler algorithm in respect to the quality of the achieved results.

5 DISCUSSION AND CONCLUSIONS

We have described an agent-based intelligent optimization framework designed to tackle several complex problems within the visualization and rendering domain.

Concurrently we have endorsed as a design goal a high level of independence between the problem solver and application, whic has been achieved intelligent optimization (e.g. metaheuristics) as the core of optimization engine.

The conducted experiments and results demonstrate the suitability of the proposed approach

As future work it is intended to extend the proposed framework with more optimization modules such as Tabu Search (Glover, 1986), Ant Colony Optimization (Dorigo and Stutzle, 2004) or Particle Swarm Optimization (Kennedy and Eberhart, 1995). A more challenging goal is to integrate capabilities to negotiate among a community of distinct optimization agents (e.g. providing distinct optimization methods) in order to obtain the best solution for a given problem or set of problems. The application of the proposed framework to other relevant problems, such as the automatic path advice (Andújar et al., 2004) leading to assisted exploration of scenes within the context of virtual environments is also envisioned.

REFERENCES

Andújar, C., Vázquez, P., and Fairén, M. (2004). Wayfinder: Guided tours through complex walkthrough models. *Computer Graphics Forum*, 23(3):499–508.

Blum, C. and Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308.

Cern, V. (1985). A thermodynamical approach to the traveling salesman problem. *Journal of Optimization Theory and Applications*, 45(1):41–51.

Certo, J., Cordeiro, N., Reinaldo, F., Reis, L. P., and Lau, N. (2006). *Advances in Artificial Intelligence*, volume vol. 26 of *Research in Computing Science*, chapter FCPx: A Tool for Evaluating Teams' Performance in RoboCup Rescue Simulation League, pages 137–148. National Polytechnic Institute.

Dorigo, M. and Stutzle, T. (2004). *Ant Colony Optimization*. MIT Press.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5):533–549.

Holland, J. H. (1975). *Adaption in natural and artificial systems*. The University of Michigan Press, Ann Harbor.

Kennedy, J. and Eberhart, R. C. (1995). Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*, pages 1942–1948.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E., and Matsubara, H. (1997). RoboCup: A challenge problem for AI. *AI Magazine*, 18(1):73–85.

Moreira, P. M., Reis, L. P., and de Sousa, A. A. (2006). Best multiple-view selection: Application to the visualization of urban rescue simulations. *IJSIMM - International Journal of Simulation Modelling*, 5(4):167–173.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS