Résolution d'un problème d'aménagement spatial à l'aide d'un algorithme génétique

Stéphane SANCHEZ, Olivier Leroux, Véronique Gaildrat, Hervé Luga
IRIT, Laboratoire de Recherche en Informatique de Toulouse
email : {sanchez, leroux, gaildrat, luga}@irit.fr

Résumé : Dans le domaine de la modélisation déclarative, la conception et la réalisation de solveurs de contraintes permet de proposer des outils performants pour la résolution des problèmes d'aménagement spatial. Les solveurs réalisés (ORANOS) ou en cours de conception (JACADI) au sein de notre équipe utilisent des méthodes basées sur un formalisme issu de l'Intelligence Artificielle : les CSP. Bien que ces méthodes complètes puissent résoudre des problèmes d'aménagement spatial sous contraintes, il peut être intéressant d'envisager de nouvelles voies de conception, notamment l'utilisation des méthodes stochastiques. C'est pourquoi, afin d'évaluer des solutions envisageables pour ce type de problème, nous proposons, au cours de cette étude, d'utiliser une méthode évolutionniste particulière (les Algorithmes Génétiques) pour la conception et la mise en oeuvre d'un nouveau solveur permettant le placement sous contraintes d'objets dans une scène tridimensionnelle.

Mots clés : Algorithmes Génétiques, Aménagement spatial, Contraintes géométriques, Solveur de contraintes, Contraintes génétiques.

1 Introduction

Le projet DEM²ONS (Declarative Multimodal MOdeliNg System) a pour but la modélisation déclarative d'environnements virtuels. Ces recherches ont abouti à la conception d'un premier solveur de contraintes [Kwa98] nommé ORANOS basé sur les DHNCSP (Dynamic Hierarchical Numeric Constraint Satisfaction Problems). Or, ORANOS présentant certaines limitations, un nouveau solveur, extension du premier est actuellement étudié : JACADI [LCG00]. JACADI étant basé sur les DHSCSP (Dynamic Hierarchical Spatial CSP), pourtant il nous a semblé intéressant d'envisager simultanément de nouvelles voies de résolution.
Nous proposons donc la conception et la mise en œuvre d'un autre solveur, non pas concurrent de JACADI mais proposant une alternative de conception, s'appuyant sur une méthode stochastique dédiée à l'optimisation et à la résolution de problèmes complexes : les Algorithmes Génétiques.

2 Problématique

Le but de cette étude est de concevoir et réaliser un solveur de contraintes géométriques pouvant résoudre un problème d'aménagement spatial particulier, le placement des meubles dans une pièce, par l'utilisation d'une méthode évolutionniste : les Algorithmes Génétiques.
Pour la conception d'un tel outil, il est nécessaire d'adapter le mode de fonctionnement des algorithmes génétiques, ainsi que la nature et la forme des contraintes géométriques propres aux problèmes d'aménagement spatial, afin de pouvoir, par la suite répondre aux trois questions suivantes :
1- Quelles sont les contraintes géométriques à retenir pour énoncer le problème ?
2- Comment modéliser l'espace de placement, les objets et les contraintes pour les adapter à une résolution par algorithme génétique ?
3- Comment évaluer les contraintes pour pouvoir noter les individus ?

Synoptique du modeleur déclaratif DEM²ONS (cf. Figure 1) : Le module de *description* se charge de décomposer les propriétés de haut niveau énoncées par l'utilisateur en un ensemble de contraintes traitables par le module de *génération*. La présence de plusieurs solveurs au sein du module de résolution se justifie par la diversité des problèmes à traiter. Aucun outil n'est suffisamment généraliste et seule la multiplication des approches et des algorithmes permet la résolution efficace d'une vaste gamme de problèmes.

Signalons encore que le couplage fort entre le modeleur classique et le modeleur déclaratif entraîne le processus inverse : toute modification directe de la scène au travers des commandes offertes par le modeleur classique doit avoir une répercussion sur le modèle du monde virtuel géré par le module de *description*.
L'interface multimodale, quant à elle, permet d'augmenter les moyens d'expression de l'utilisateur, en lui offrant de nombreuses possibilités d'interaction.

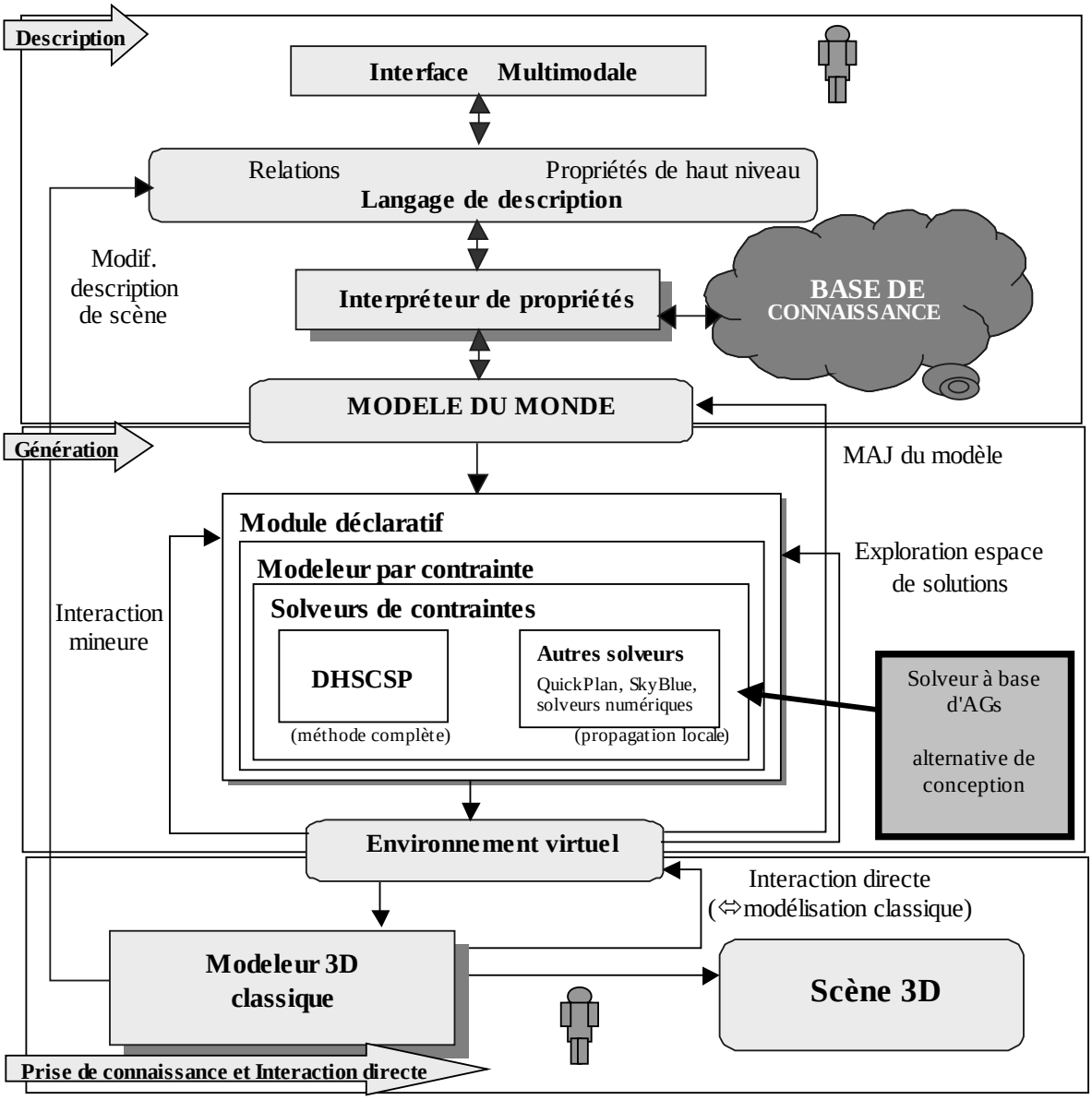


Figure 1: positionnement de notre étude dans le projet DEM²ONS

3 Les Algorithmes Génétiques (AGs) :

Koza [Koz92] donne une bonne définition des algorithmes génétiques :
Un algorithme génétique est un algorithme mathématique fortement parallèle qui transforme un ensemble (une *population* d'individus sous forme d'objets mathématiques (généralement des chaînes de caractères de longueur fixe assemblées en *chromosome*), chaque individu étant associé à une valeur d'évaluation (*fitness*)) en une nouvelle population (i.e., la *génération* suivante) par l'utilisation d'opérations issues des principes Darwiniens de reproduction et de survie de l'individu le mieux adapté et des opérations génétiques en découlant.

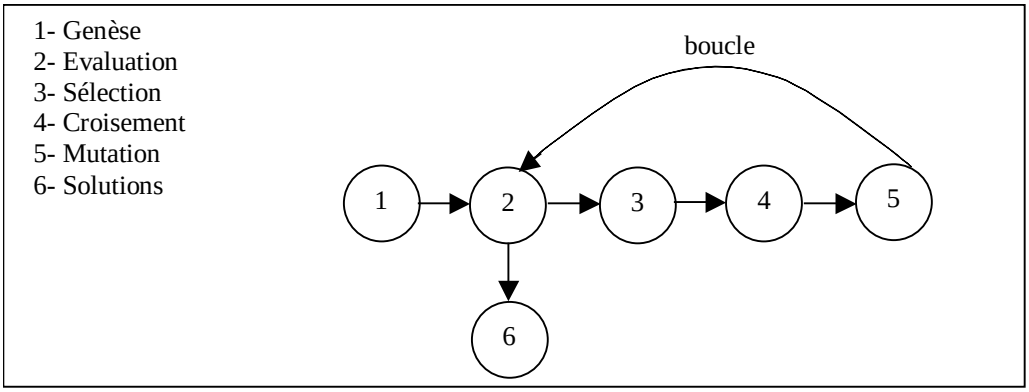


Figure 2 : déroulement général d'un AG

4 Le solveur de contraintes géométriques :

4.1 Caractéristiques :

La définition des caractéristiques envisagées pour le solveur expérimental à base d'algorithmes génétiques sont similaires à celles spécifiées pour JACADI [Ler99]. Ceci permettra de proposer le nouveau solveur comme alternative de conception pour les cas où la méthode de résolution utilisée par JACADI ne pourrait être appliquée (en particulier dans le cas d'une scène comprenant un grand nombre d'objets dont la présence ou la position n'est pas critique).

Dans le cadre de cette étude sur l'utilisation des Algorithmes Génétiques pour la résolution de problèmes d'aménagement spatial, nous avons défini pour le solveur certaines caractéristiques :
- Résolution des types de contraintes géométriques suivantes: contraintes topologiques ("l'armoire est contre le mur"), contraintes métriques ("la chaise se trouve à un mètre devant le bureau"), contraintes angulaires ("disposer trois chaises uniformément autour de la table ronde"), contraintes de sens ("la chaise est en face du canapé"), contraintes de dimension : ("la table fait 1,80m de long").
- Représentation réaliste du monde modélisé : espace de placement et objets en 3 dimensions, orientations quelconques, respect du non-recouvrement des objets et de l'équilibre.
- Gestion de contraintes hiérarchiques afin de pouvoir exprimer des préférences entre les contraintes ou résoudre les cas sur-contraints[1].
- Gestion des graphes de contraintes cycliques.
- Possibilité de contraintes disjonctives ("mettre les chaises contre UN mur", "placer la poubelle à droite OU à gauche du bureau") et la négation ("l'armoire N'est PAS devant la fenêtre")

4.2 Intérêt de l'utilisation des Algorithmes Génétiques :

Les méthodes d'optimisation par algorithmes génétiques sont connues pour donner d'excellents résultats aussi bien dans les petits que dans les vastes espaces de recherche. Donc, concevoir un solveur de contraintes à base d'algorithmes génétiques devrait permettre d'obtenir un outil capable de résoudre des problèmes difficilement solubles par des techniques issues des CSP.
Un problème sur-contraint n'est pas une gêne pour un algorithme génétique : quelque soit le nombre de contraintes non satisfaites, l'individu le mieux noté, donc la meilleure solution potentielle, sera toujours celui qui satisfait au mieux les contraintes.
La hiérarchisation des contraintes est facilement réalisable: l'ajout d'un poids supérieur sur une contrainte permet d'accroître la pression qu'elle effectue sur la population (ceci augmente l'apport de sa note au niveau de l'évaluation des individus). Toutefois, l'AG cherchant à produire les individus de

[1] Un problème est dit *sur-contraint* quand il n'existe pas de solution. En général la cause du problème vient d'un ou plusieurs sous-ensembles de contraintes qui ne peuvent être satisfaits simultanément.

meilleure qualité possible, il n'abandonnera les contraintes de poids inférieur que si elles ne peuvent en aucun cas être satisfaites en même tant que les contraintes prioritaires (cas des problèmes sur-contraints).
Dans le cas des algorithmes génétiques l'implémentation d'un problème est essentiellement descriptive: il faut définir des modèles évaluables des attributs du problème et la forme des solutions recherchées afin de pouvoir le résoudre.
La modélisation du problème à résoudre (l'aménagement d'une scène dont les objets à placer sont les individus potentiellement solutions) et les contraintes (les attributs du problème) sera réalisable par une description naturelle du monde réel . Par exemple, les meubles pourront être représentés par leur positon, leur orientation et leurs dimensions. La plus grande difficulté sera alors de concevoir les fonctions d'évaluation des différentes contraintes géométriques.
Cependant, l'utilisation d'un algorithme génétique présente un inconvénient : étant une méthode stochastique et, donc non complète, elle ne permet pas de fournir l'ensemble complet des solutions du problème. Au maximum on pourrait atteindre autant de solutions que d'individus de la population mais, en raison du mécanisme de convergence (basé sur la propagation des meilleures caractéristiques de chaque individu), cette dernière est très homogène et, donc il y a de fortes chances que plusieurs solutions soient identiques.

4.3 Représentation de l'espace de placement, des objets et des contraintes :

Cette phase de modélisation est essentielle pour la mise en œuvre du solveur : l'espace de placement et les objets de la scène définissent l'environnement de la population l'algorithme génétique, les contraintes représentent la pression de cet environnement sur les individus de la population (création d'une fonction de *"fitness"* permettant leur évaluation), enfin les objets à placer représenteront les gènes des individus.

L'espace de placement, dans lequel seront situés les objets, est en trois dimensions, orthonormé (axes x, y, z, le plan (xy) représentant le sol), orienté en radians selon l'axe z, borné, et doté de points cardinaux.
Les objets de la scène : afin de prendre en compte des objets réalistes, ils sont représentés par une hiérarchie de boites englobantes. Ceci permet la modélisation d'objets avec différents niveaux de détails (une armoire fermée sera définie par une seule boite englobante, une chaise par une hiérarchie détaillant l'assise, le dossier et les quatre pieds). Cela offre la possibilité de contraintes réalistes (poser un objet sur une chaise, se fait généralement sur l'assise).
Chaque boite englobante est nommée et définie par : un point de référence, son orientation selon l'axe z (entre 0 et 2π avec un pas de $\pi/8$), ses dimensions en x, y et z. Par défaut, les seuls objets imposés dans la scène sont les 4 murs de la pièce (correspondant aux frontières de l'espace de travail) et le sol.
L'utilisation de valeurs entières ou discrètes a pour but de limiter l'espace de recherche des solutions, et donc de favoriser la convergence de l'algorithme génétique constituant le noyau du solveur, sans toutefois trop dégrader la qualité des solutions proposées.
Les contraintes : les contraintes utilisées par le solveur sont des contraintes primitives (de bas niveau d'abstraction). En les combinant, l'utilisateur énonce un problème à soumettre au solveur. Ces contraintes primitives doivent, par leur conception, permettre au solveur de proposer, si elle existe, au moins une solution cohérente et réaliste au problème posé.
Afin de satisfaire ces exigences, deux catégories de contraintes primitives sont utilisées : les contraintes physiques et les contraintes géométriques.
Les contraintes physiques ont pour but de forcer le solveur à présenter à l'utilisateur des solutions réalistes : une contrainte de *non-recouvrement* permet d'invalider tout objet en collision avec un autre, une contrainte de *gravité* invalide un objet non supporté par le sol ou un autre objet présent dans la scène. Les contraintes géométriques permettent d'énoncer par combinaisons booléennes (opérateurs ET, OU, NON) le problème de placement à résoudre.
Trois types de contraintes géométriques sont utilisés : des contraintes de *zone* qui permettent de définir une région de l'espace de placement par rapport à un objet de référence ("placer l'objet A *à droite* de l'objet B"), des contraintes d'*orientation* ("placer l'objet A face à l'objet B") et des contraintes de *distance* ("placer l'objet A *à 1 mètre* de l'objet B").

Chacune de ces contraintes primitives est modélisée par un ensemble de propriétés mathématiques et géométriques simples. Par exemple, une contrainte de *zone* est représentée par un volume défini par un polygone de contrôle dans le plan xy et un intervalle de hauteur autorisées (coordonnées en z) :

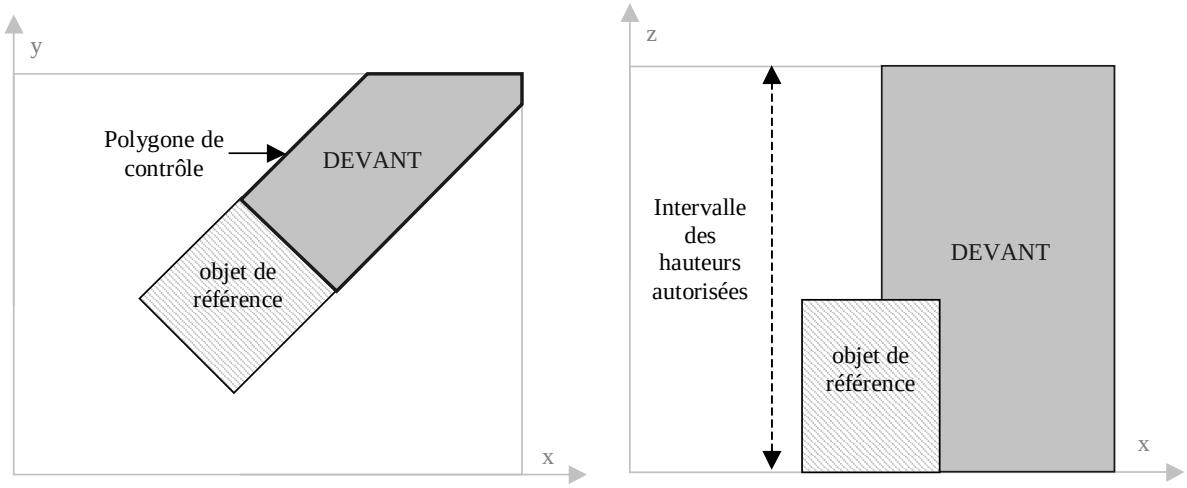


Figure 3 : Volume représentant la contrainte de zone DEVANT

4.4 La résolution du problème de placement par un AG :

Enoncé du problème : la formulation du problème se fait en deux étapes : la construction du génome d'un individu type qui représente les objets à placer dans la scène (on fournit la forme des solutions au solveur) et la construction d'un vecteur de contraintes à appliquer aux objets à placer (on construit la fonction d'évaluation des individus). Ce couple "individu type / vecteur de contraintes" définit exactement le problème que le solveur doit résoudre.
- Construction du génome : Un individu représente pour le solveur une solution potentielle au problème posé. Chaque individu est caractérisé par son génome. L'évaluation du génome permet d'attribuer une note à l'individu. Cette note fournit une estimation de la qualité de l'adaptation de l'individu face au problème à résoudre. Le génome d'un individu est formé de chromosomes. Chaque chromosome correspond à une série d'objets à placer selon une combinaison de contraintes lui correspondant. Chaque gène du chromosome correspond à un seul objet. Le gène contient les paramètres de l'objet à placer. L'utilisateur doit construire le génome d'un individu afin de présenter au solveur le nombre de séries d'objets à placer simultanément, ainsi que le nombre et les dimensions des objets par série. Tous les objets d'une série ont les mêmes dimensions. Les dimensions des objets sont soit des valeurs entières fixes, soit des intervalles de valeurs entières autorisées (pseudo-contraintes de dimension).
- Le vecteur de contraintes contient des combinaisons booléennes de contraintes. Afin d'associer chaque combinaison de contraintes à une série précise d'objets à placer, le vecteur doit avoir la même structure que le génome de l'individu précédemment créé. Ainsi, si on désire placer N séries d'objets, ce qui correspond à un individu dont le génome à N chromosomes, alors le vecteur de contraintes contiendra N combinaisons, une par série. A ce niveau, l'utilisateur doit, si nécessaire, fixer le poids (priorité) de chaque contrainte élémentaire.

Evaluation des individus : Cette étape est la plus importante. En effet, la convergence de l'algorithme vers une solution va dépendre exclusivement de la qualité de l'évaluation des individus : la note donnée à un individu doit permettre de traduire, le plus exactement possible, le degré de satisfaction aux contraintes qui lui sont appliquées. De plus, les méthodes évaluant l'individu par rapport aux contraintes élémentaires doivent donner des résultats homogènes afin de pouvoir les combiner sans que le système crée une hiérarchie de contraintes non désirée (on utilisera un pourcentage de satisfaction). Enfin, il faut que les algorithmes utilisés par ces méthodes soient assez rapides pour que le solveur ait un temps de réponse acceptable.
L'évaluation d'un individu se fait en trois étapes : évaluation des gènes de l'individu par rapport aux contraintes physiques pour déterminer leur validité, puis évaluation des gènes par rapport aux

contraintes qui leur sont associées, les gènes invalides ne sont pas évalués et enfin mise en commun des résultats des évaluations pour le calcul de la note finale de l'individu.
L'évaluation de la satisfaction des différentes contraintes géométriques est fait par l'utilisation d'un pourcentage de satisfaction permet la convergence de l'algorithme.
Par exemple pour la contrainte de *zone* énoncée positivement l'évaluation se résume comme suit :

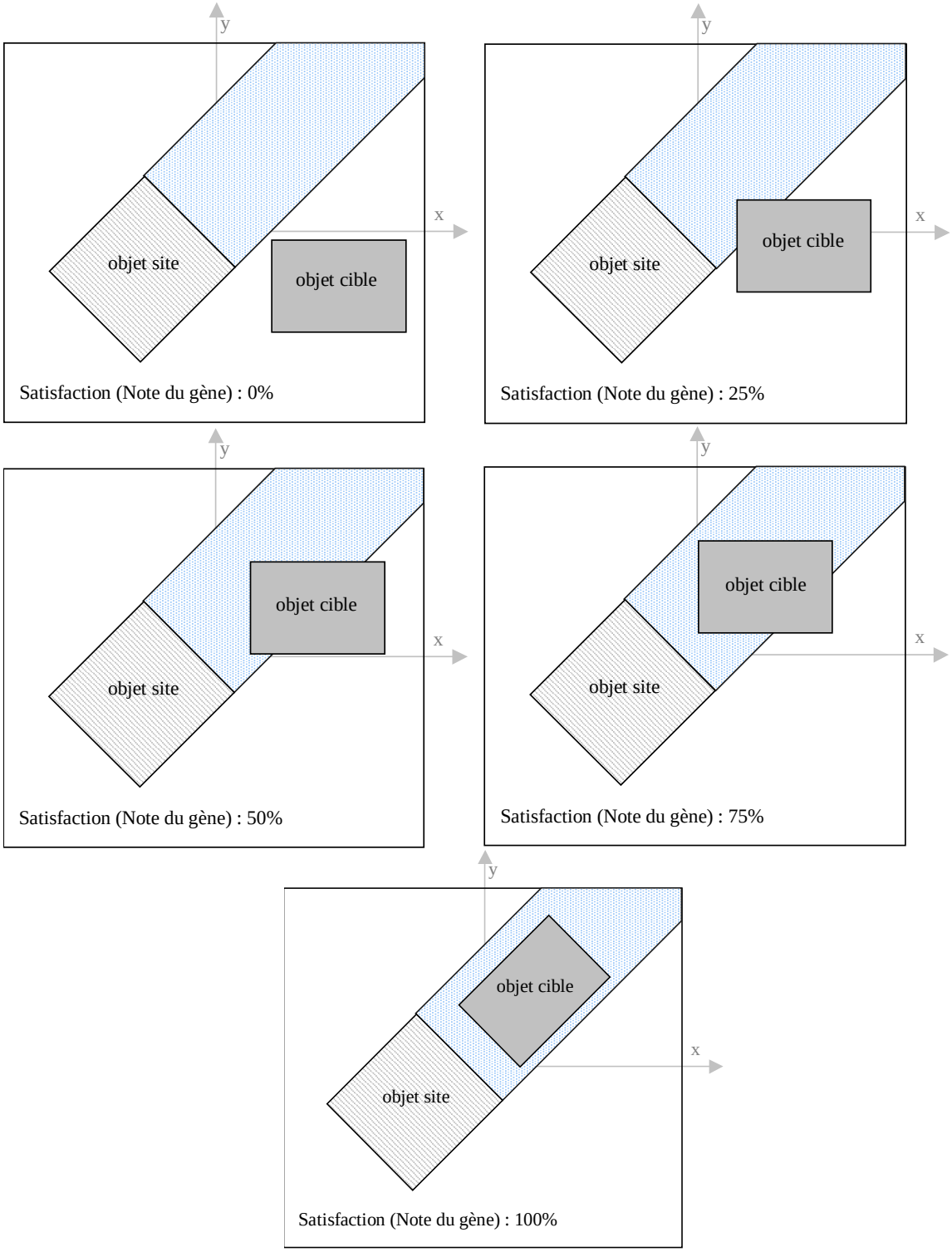


Figure 4 : convergence d'une contrainte de zone

Après évaluation d'un gène d'un individu, on obtient une note par contrainte associée au chromosome auquel il appartient. La note finale d'un individu est alors calculée en deux étapes : le calcul des notes de chaque chromosome puis la mise en commun de ces notes.

Le calcul des notes des chromosomes se fait par somme pondérée afin de prendre en compte la hiérarchie des contraintes :

$$noteChromo_i = \frac{\sum_{k=1}^{nConts_i} \sum_{j=1}^{nGenes_i} prio_k \times noteGC_{ijk}}{nGenes_i \times \sum_{k=1}^{nConts_i} prio_k}$$

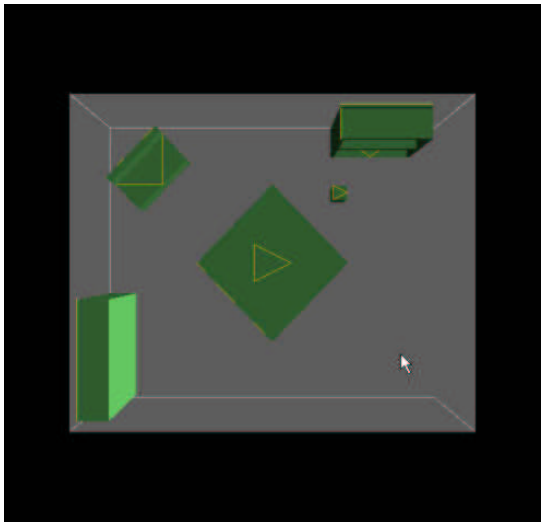
Avec : $noteChromo_i$: note du chromosome i de l'individu en cours d'évaluation.
$noteGene_{ij}$: note du gène j du chromosome i.
$noteCont_{ik}$: note correspondant à la satisfaction de la contrainte k par l'ensemble des gènes du chromosome i.
$noteGC_{ijk}$: note du gène j du chromosome i obtenue pour son évaluation par rapport à la contrainte k.
$nGenes_i$: nombre de gènes du chromosome i.
$nConts_i$: nombre de contraintes associés au chromosome i.
$prio_k$: priorité de la contrainte k.

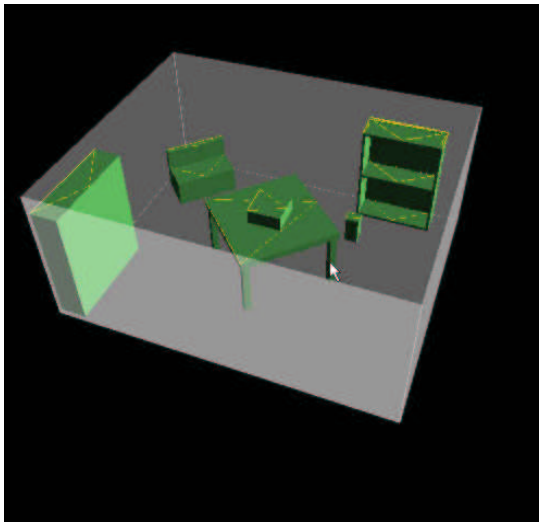
Au cours de cette étude sur la réalisation d'un solveur de contraintes à base d'algorithme génétique, nous n'avons pas envisagé d'utiliser des méthodes multi-objectifs, c'est pourquoi la note finale est calculée simplement comme une moyenne des notes des chromosomes de l'individu :
Soient : *noteIndividu* : la note finale de l'individu.
nChromos : le nombre de chromosomes du génome de l'individu.
Le calcul de la note finale est alors:

$$noteIndividu = \frac{\sum_{i=1}^{nChromos} noteChromo}{nChromos}$$

5 Résultats

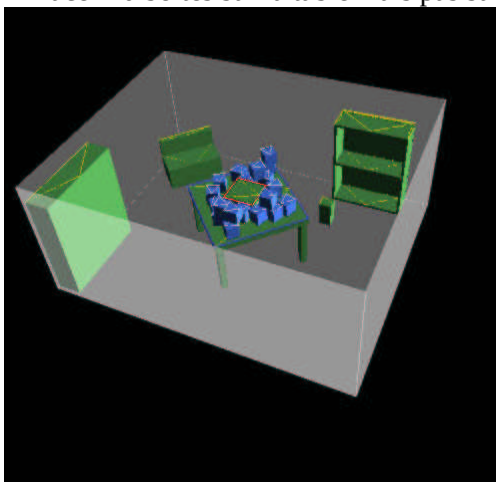
L'espace de placement : L'espace de travail choisi est une pièce rectangulaire de 5 mètres par 4 et de 2 mètres de hauteur. Les objets présents dans la scène sont tous définis par des boites englobantes hiérarchiques et ils ne sont pas forcément parallèles aux murs de la pièce.
Ces objets sont : une table composée d'un plateau et 4 pieds, une armoire fermée, un fauteuil composé d'une assise et d'un dossier, un tabouret, une boîte posée sur la table, une étagère composée d'un fond, de deux montants et de trois rayons.





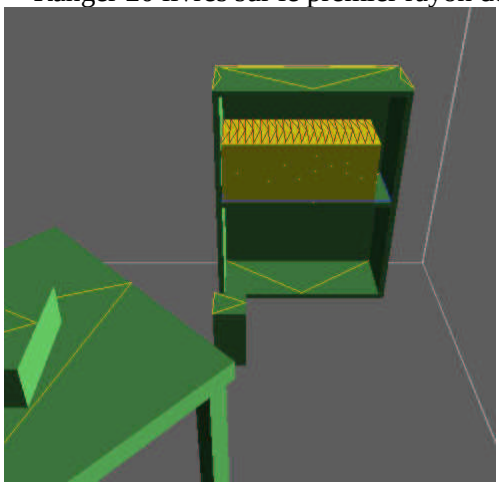
Images résultats et temps de réponse : Les temps de réponse moyens fournis sont calculés sur dix résolutions consécutives sur un Celeron 450 Mhz.

- « Placer 20 boites sur la table mais pas sur la boite déjà posé sur la table »



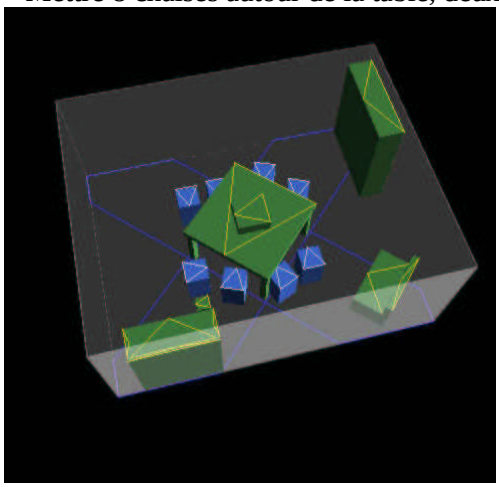
Temps de réponse moyen : 3 min 14,990 s

- « Ranger 20 livres sur le premier rayon de l'étagère »



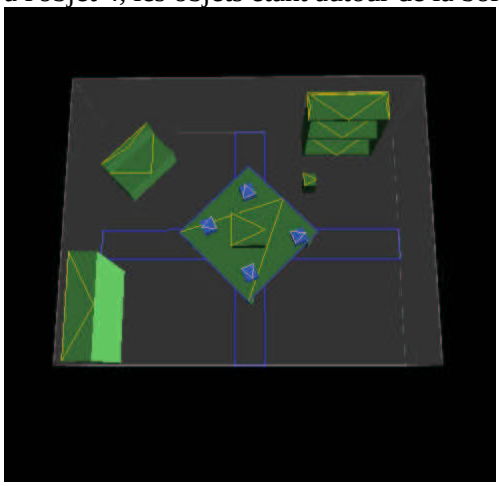
Temps de réponse moyen : 3 min 55,010 s

- « Mettre 8 chaises autour de la table, deux par côtés et dirigées plus ou moins vers la table »



Temps de réponse moyen : 43,698 s

- « 4 objets sur la table l'objet 2 orienté à 90° par rapport à l'objet 1, l'objet 3 orienté à 90° par rapport à l'objet 2, l'objet 4 orienté à 90° par rapport à l'objet 3 et l'objet 1 orienté à 90° par rapport à l'objet 4, les objets étant autour de la boite posée sur la table »



Temps de réponse moyen : 6,286 s

6 Conclusion

Pour une première étude sur l'utilisation d'un solveur de contraintes par Algorithme génétique, les résultats obtenus semblent prometteurs :
- la base de primitives retenues a permis d'énoncer tous les problèmes simples ("sur", "contre", "face", "au-dessus", "dans", "parallèle à",...), et la plupart des problèmes plus complexes envisageables en combinant les contraintes. Les seuls énoncés que nous n'avons pas pu produire avec le solveur actuel sont essentiellement certaines combinaisons hiérarchiques de contraintes utilisant l'opérateur OU. Ceci n'est pas dû à un défaut de la méthode utilisée, mais au fait que la gestion des opérateurs booléens n'est pas encore finalisée.
- Tous les problèmes non sur-contraints que nous avons pu énoncer ont été résolus.
- Pour les problèmes sur-contraints, le solveur a toujours atteint la meilleure solution possible en fonction des priorités de chaque contrainte. Certains problèmes ont même été volontairement énoncés sur-contraints afin d'orienter la convergence vers un type particulier de solutions.
- Les seuls cas de non-convergence du solveur (notes des individus bloquées à 0), et donc d'échec de la résolution, sont les problèmes qui n'ont aucune solution possible (par exemple, "placer la chaise devant ET derrière la table").
- Les problèmes générant un graphe de contraintes cyclique ont été traités sans aucun problème. Le solveur ne fait pas de différence : pour lui, il faut seulement optimiser la note des individus par rapport aux contraintes.
- La négation est gérée par le solveur comme une contrainte classique et n'a donc posé aucune difficulté de résolution.
- Les contraintes hiérarchiques sont traitées correctement et, lorsque cela est possible, la forme de la solution est celle recherchée.

Cependant, le solveur proposé n'est pas encore au point et, bien que proposant déjà certaines améliorations par rapport à ORANOS (orientation des objets quelconques, espace de placement en dimensions, gestion des graphes de contraintes cycliques, prise en compte de la négation), il présente encore des défauts :

- Les temps de réponse d'un problème à l'autre sont très variables et ne correspondent pas directement au nombre d'objets à placer ou au nombre de contraintes à traiter : une étude sur la convergence du solveur semble nécessaire.
- L'évaluation par somme pondérée ne permet pas au solveur de présenter un ensemble de solutions. Néanmoins, l'utilisation de méthodes d'évaluation multicritères de type MOGA [FF95b] semblent pouvoir corriger, du moins partiellement, ce défaut.

7 Bibliographie

[BS00] Stéphane Sanchez, Alain Berro. Optimisation par Algorithme Génétique du placement des succursales d'une entreprise. ROADEF 2000 (Recherche Opérationnelle et Aide à la DEcision Française), Nantes, Janvier 2000.

[CC98] C.A.C. Coello, A.D. Christiansen. Two new GA-based methods for multi-objective optimization . Civil Engineering Systems. (In Press), 1998.

[Cha95] P. Charman. Gestion des contraintes géométriques pour l'aide à l'aménagement spatial. Thèse de Doctorat, Ecole nationale des Ponts et Chaussées, Novembre 1995.

[Dar59] C. Darwin. The Origin of Species by Means of Natural Selection or the Preservation of Favoured Races in the Struggle for Life, 1859.

[FF93] C.M. Fonseca, P.J. Fleming. Genetic Algorithms for Multi-objective optimization : formulation, discussion, and generalization. In Stephanie Forrest, editor, Proceedings of the Fifth International Conference On Genetic Algorithms, pp. 416-423. Morgan Kaufmann , 1993.

[FF95a] C.M. Fonseca, P.J. Fleming. Multi-objective Optimization and Multiple Constraint Handling with Evolutionary Algorithms I : A Unified Formulation. Technical Report 564 (January), University of Sheffield, Sheffield U.K., 1995.

[FF95b] C.M. Fonseca, P.J. Fleming. An Overview of Evolutionary Algorithms in Multi-objective Optimization. Evolutionary Computation 3, 1 (Spring), 1-16. 1995.

[Gol89] D.E. Goldberg. Genetic algorithms for search, optimization, and machine learning. Addison-Wesley, Reading, Massachusetts, 1989.

[HNG94] J. Horn, N. Nafpliotis, D.E. Goldberg. A niched Pareto genetic algorithms for multi-objective optimization. In Proceedings of the ICEC'94, IEEE, pp. 82-87, 1994.

[Hol92] J.H. Holland. Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence (Second ed.). MIT Press, Cambridge, Massachusetts, 1992.

[KGC97] G. Kwaiter, V. Gaildrat, R. Caubet. DEM²ONS : A High Level Declarative Modeler for 3D Graphics Applications. In Proceedings of the International Conference on Imaging Science Systems and Technology, CISST'97, pages 149-154, Las Vegas, June 30-July 3, 1997.

[KGC98a] G. Kwaiter, V. Gaildrat, R. Caubet. Controlling Objects Natural Behaviors with 3D Declarative Modeler. In Proceeding of Computer Graphics International, CGI'98, Hanover, Germany, pages 248-253, 24-26 June 1998.

[KGC98b] G. Kwaiter, V. Gaildrat, R. Caubet. Modelling with Constraints : A Bibliographical Survey. In Proceeding of International Conference on Information Visualization, IV'98, London UK, 29-31 July, 1998.

[Koz92] J.R. Koza. Genetic Programming. On the Programming of Computers by Means of Natural Selection. The Mit Press, 1992.

[Kwa98] G. Kwaiter. Modélisation déclarative de scènes : étude et réalisation de solveurs de contraintes. Thèse de Doctorat, Université Paul Sabatier, Toulouse, Décembre 1998.

[Ler99] O. Le Roux. Solveur de contraintes pour la modélisation déclarative : Etude du problème de l'aménagement spatial de scènes tridimensionnelles sous contraintes. DEA IIL, Université Paul Sabatier, Toulouse, 1999.

[LGC00] O. Leroux, V. Gaildrat, R. Caubet. Design of a New Constraint Solver for 3D Declarative Modelling: JACADI, In Proceedings of 3IA, Limoges, mai 2000.

[Lug97] H. Luga. Vie artificielle et Synthèse d'image : Etude des mécanismes évolutionnistes pour la synthèse de formes et de comportements. Thèse de Doctorat, Université Paul Sabatier, Toulouse, 1997.

[Par96] V. Pareto. Cours D'Economie Politique, Volume I et II. F. Rouge, Lausanne, 1896.

[Sch91] J.D. Schaffer. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In Grefenstette, J.J., editor, Proc. First Int. Conf. On Genetic Algorithms, pp.93-100. Lawrence Earlbaum, 1991.

[Ste86] R.E. Steuer. Multiple criteria optimization : Theory, computation, and application. John Wiley, New York,