Massively Parallel Genetic Algorithm for Physically Correct Articulated Figure Locomotion

J. Thomas Ngo
Committee on Higher Degrees in Biophysics
Harvard University
ngo@tammy.harvard.edu

Joe Marks
Cambridge Research Lab
Digital Equipment Corporation
marks@crl.dec.com

1 Introduction

Increasing physical accuracy has been a trend in many areas within the field of computer graphics. In this paper we discuss our approach to a problem that arises in the design of physically realistic animations that feature autonomous characters modeled as articulated figures. The problem is to specify a character's trajectory (*i.e.*, to compute the time-dependent coordinates of each figure's parts) so that the resulting motion is physically correct and conforms to the animator's goals for the character (Badler et al., 1991).

Witkin and Kass (1988) captured the requirements of this problem in the Spacetime Constraints (SC) paradigm. Attention is focused on minimizing the responsibility of the animator to specify details of the motion, leaving only the following specifications to human choice:

- the character's physical structure and the actuators it may use to effect motion;
- the overall goal to be accomplished; and
- conditions on the nature of the motion.

The remaining task, which is delegated to the computer, is to compute a trajectory for the character that is consistent both with physical law and with the given specifications. For example, given the goal of achieving maximal horizontal progress in a given time, the task might be to discover how to make an animated character walk or run. This is the Spacetime Constraints problem.

Automated global solution of the general SC problem is hindered by a number of factors:

- Multimodality—For any given SC problem, there may be several, quite different, families of viable solutions.[1] In almost all cases there are local optima within each family, as well as optima that are not close to viable solutions. Analysis via the theory of NP-completeness (Garey and Johnson, 1979) demonstrates that this multimodality is likely to lead to intractability (Hopcroft et al., 1984).

- Lack of differentiability—An infinitesimal change in the behavior of an articulated linkage can sometimes lead to a large change in the evaluation function (Ngo and Marks, 1992). Some parts of the evaluation space may be differentiable, but in most cases discovery of new locomotive strategies corresponds to optimization across discontinuities.

[1]A famous example of this multimodality comes from the track-and-field high jump, a familiar task involving human characters that can be posed as an SC problem. First introduced at the 1968 Olympic Games, the "Fosbury Flop," a technique that involves jumping backwards over the bar, has replaced the previously standard Western roll and straddle as the method of choice for world-class high jumpers. An even earlier technique, the scissors, or Eastern, method was dropped when cushioning material was allowed in the landing pit.

- Stiffness—The amount of CPU time required to evaluate a single candidate solution has a low theoretical complexity, but is large in absolute terms. This is because the equations of motion are stiff: accurate simulation must be done on a fine timescale, whereas interesting behavior occurs on a longer timescale.

The informal statement of the SC problem given above is very general. To develop useful algorithms, we must consider restricted, more specific versions of the problem. We have considered a form of the SC problem that contains a number of simplifications:

- The system is simulated in two dimensions only.
- The only external object is the ground, which must be horizontal.
- The initial conditions consist of positional constraints only and must be specified fully.
- The muscle torques that can be applied about the hinges in the linkage are not subject to hard constraints. (However, terms in the evaluation function might be used to restrain excessive torques.)

Although a much restricted version of the general Spacetime Constraints problem, this statement includes all of the difficulties listed above. The simplifications incorporated therein do not necesssarily reflect inherent limitations of our algorithm (for example, the extension to non-zero initial velocities is trivial); rather, we have chosen to provide the most economical problem description that covers the SC problem instances we have considered.

2 Algorithm

We have implemented a parallel genetic algorithm (GA) using the C* language on a Thinking Machines CM-2 with 4096 processors.

GAs are attractive prospects for the SC problem for a variety of reasons (Holland, 1975). While not typically the fastest solutions to a given problem, GAs are sometimes the most robust because they cope well with multimodality. They do not require gradient information, and are therefore not limited by evaluation-space discontinuities encountered in SC problems. Finally, they appear well suited for implementation on parallel hardware.

The principal challenge in designing a GA for use in a particular problem domain is to select an appropriate underlying representation. Although considerable effort has been spent on attempts to develop a universal, bit-based GA, in many cases practical efficiency is achieved only if crossover and mutation operators—and hence the underlying representation—can be tailored for a given problem (Davis, 1991).

The need for problem-specific treatment has been of particular importance in the case of physically realistic trajectory planning. Time-domain representations are natural for local optimization (Witkin and Kass, 1988; Brotman and Netravali, 1988) because they lend themselves to perturbational analysis. However, they appear to be inappropriate for genetic solution of the global SC problem (Ngo and Marks, 1992).

We have developed a representation for the SC problem that is based loosely on a stimulus-response model (Skinner, 1938). A choice of parameters in this representation may be thought of as defining a hard-wired "brain" for a linkage; the task of the GA is to breed brains for linkages of a given structure, selecting for brain configurations that lead to behavior that best conforms to the animator's goals. Details of our stimulus-response representation are described elsewhere (Ngo and Marks, 1992). Here we define the essential concepts and illustrate the representation by describing a simple SC problem cast in anthropomorphic terms.

The form of the representation directly reflects two facts about linkage locomotion: firstly, that a linkage can affect its absolute

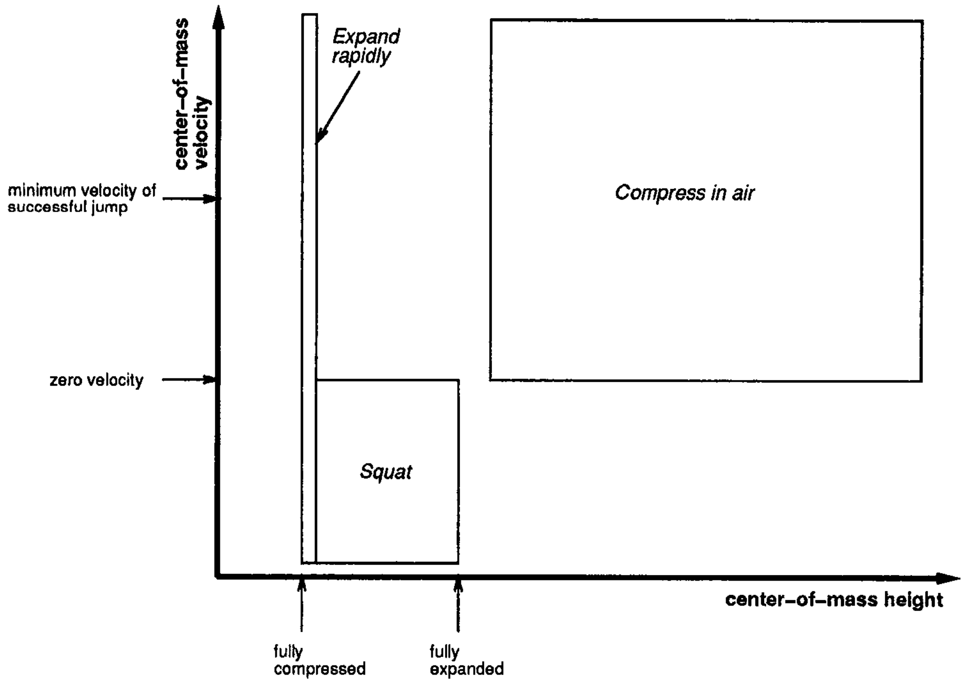

Figure 1: Manually constructed stimulus-response solution

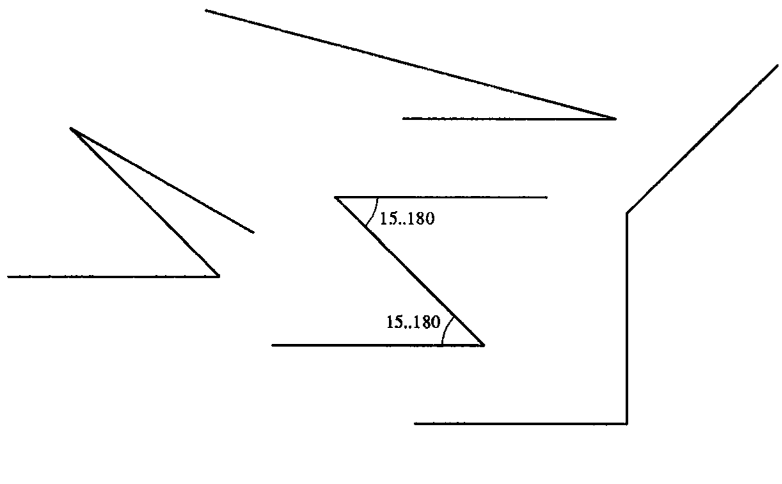

Figure 2: Structural properties of Willy Worm. All three rods are of equal mass. The center configuration is labeled with joint-angle ranges (in degrees). All of the sample configurations depicted are consistent with these joint-angle ranges.

position and orientation only by changing its internal configuration; and secondly, that most such changes in shape have consequences that depend on circumstances in the environment—for example, pushing on the ground can produce upward acceleration only if the linkage is touching the ground.

A candidate solution consists of a small set of *stimulus-response pairs*. Each pair is a *response* (a low-level prescription for changing the linkage's internal configuration in a continuous manner) and a *stimulus function* (which encodes the conditions under which the response should be executed). Each stimulus function is a scalar function in *sense space*, the space spanned by a set of predefined *sense variables*. A sense variable, in turn, may be any real-valued function of the linkage's physical state: typically, the set includes all internal joint angles, the pressure exerted by each joint on the ground, and the position and velocity of the center of mass. At each instant in its lifetime, the linkage chooses from among its repertoire of responses based on the values of the stimulus functions.

In the following anthropomorphic example, the task is for a person to jump as high as possible from a standing position, with jump height defined to be the maximum height cleared. Figure 1 depicts a good stimulus-response solution to this problem. The two sense variables are a minimal set chosen for the sake of illustration: the vertical position and velocity of the person's center of mass. Each of the three rectangles represents the region over which a particular stimulus function dominates. Corresponding to each of these regions is an associated response:

- Expand—If the person's center of mass is low, then expanding rapidly will propel him into the air.
- Squat—If the person's center of mass is too high for expansion to generate enough vertical momentum, he should first squat.
- Compress in air—If the person is moving upward and his center of mass is too high for him to touch the ground, then it is too late for him to influence the path of his center of mass. He can, however, increase the height cleared by contracting.

In constructing this simple example, we have identified appropriate stimuli (regions of sense space), actions (internal target configurations to adopt), and an appropriate mapping from the set of sense-space regions onto the set of actions. Our GA performs these steps automatically for 2D SC problems.

3 Results

Our algorithm has solved problems that appear to be beyond the grasp of existing techniques in animation because they have evaluation functions that are multimodal and discontinuous. We have presented detailed results elsewhere (Ngo and Marks, 1992). Here we describe simple, representative locomotive strategies for "Willy Worm" (Figure 2), a small but flexible 3-rod linkage that was designed for richness of behavior. In this problem, Willy is initially at rest in a Z-shaped configuration and attempts to move his center of mass as far horizontally as possible within a fixed time.

GAs cope with multimodality by allocating increasing amounts of processing time to promising areas of the search space in a *gradual* manner, rather than by pruning mediocre solutions immediately. This effect is particularly easy to observe in our variant of the GA because the individuals in the population are spread out in fixed locations on an imaginary two-dimensional torus. Mating can occur only between individuals that are close together on the torus, so relatively homogeneous colonies of similar solutions tend to form.

Different colonies often correspond to qualitatively different locomotive strategies. Figures 3 and 4 illustrate two modes of locomotion that competed in a single run. The two behaviors, which we call flipping and shuffling, are

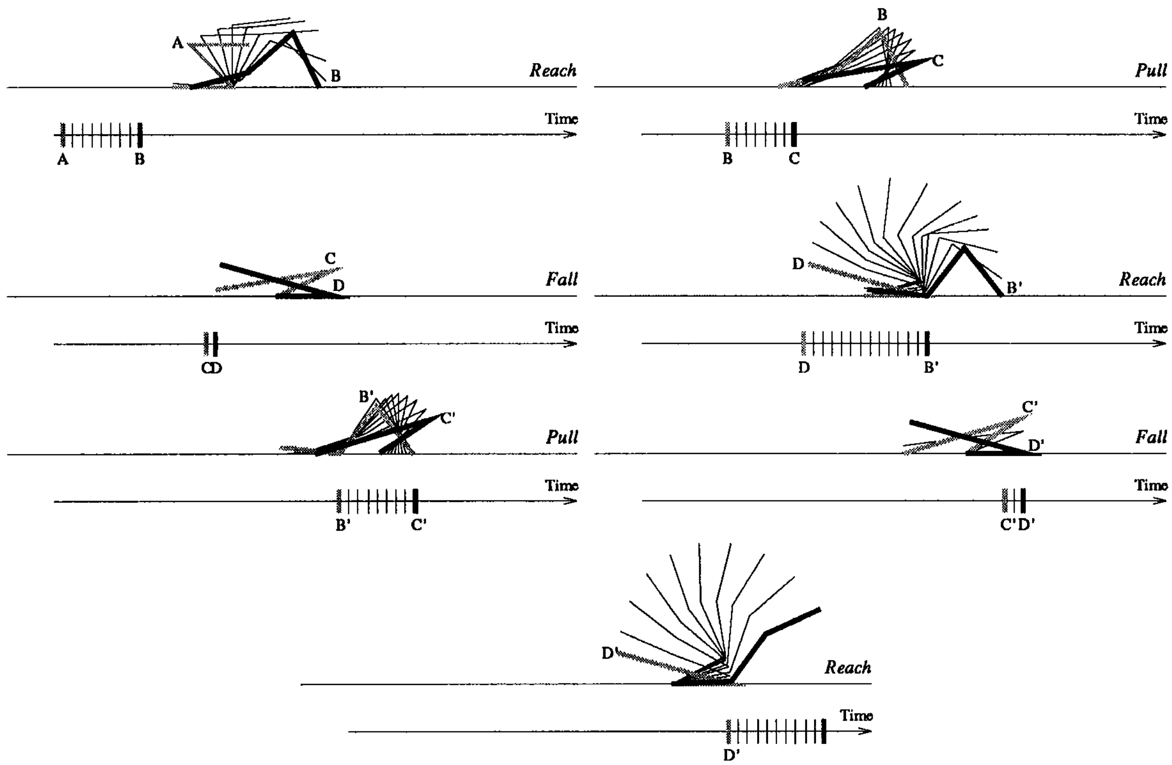

Figure 3: Willy Worm walking forward by flipping.

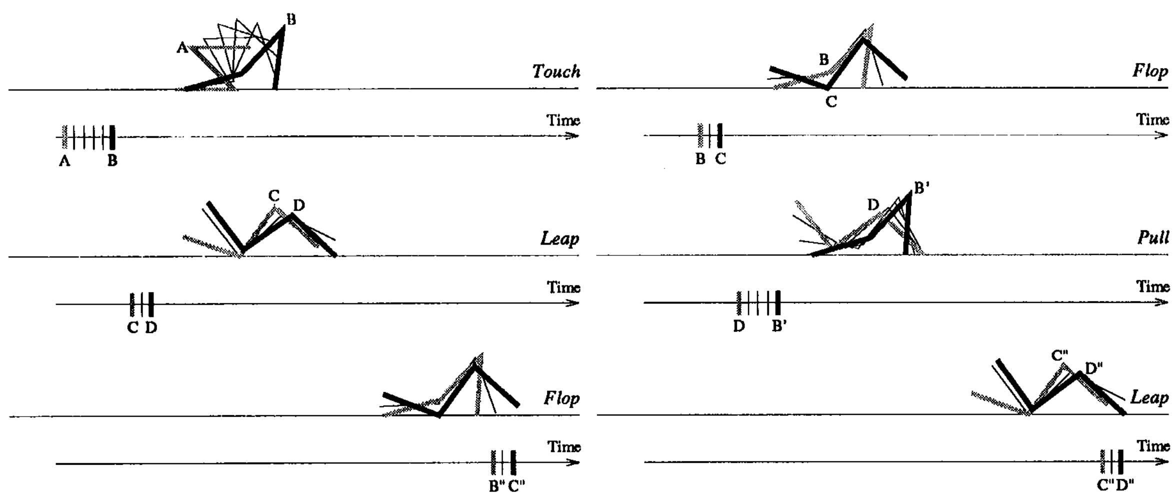

Figure 4: Willy Worm walking forward by shuffling. The full trajectory contains six short cycles similar to B–C–D–B′; for clarity we depict only the first and the last.

quite different from each other.[2] In principle, the large population size (4096) is capable of accommodating rich diversity, but because our current facilities permit recording and viewing of only the best trajectory in a given generation, we can see only the tip of the iceberg. Nonetheless, it is common for half a dozen distinct behaviors to be found among the solutions recorded in a single run.

4 Discussion

Our algorithm computes original and effective solutions to the restricted SC problems we have considered so far. The need for expertise on the part of the user has been eliminated from nearly every facet of the algorithm. In particular, the user need not be able to construct a coarse first guess at the form of the trajectory. To the best of our knowledge, our algorithm is unique in this respect. But is the success of the algorithm due to massive parallelism, or in spite of it?

Some of the key concepts in our approach are clearly independent of computer architecture. Our two most important choices—using a stochastic global search strategy in place of the usual gradient-descent methods (Witkin and Kass, 1988; Brotman and Netravali, 1988), and using a stimulus-response representation—had nothing to do with massive parallelism. These ideas would carry over directly to a serial variant of our algorithm. The other important choice we made—the use of a GA as the search engine—is the one that led us to consider massive parallelism. At first blush, the match between the GA paradigm and current incarnations of SIMD massive parallelism seems to be excellent:

- With one candidate solution per processor, and almost identical processing required for each solution, it appears possible to keep all the processors busy most of the time.
- Only local communication between processors is necessary, obviating the need for expensive global communication in the processor network.
- The simple and elegant mapping of the GA paradigm onto the CM-2 architecture makes for easy development and debugging.

Thus we concluded that the CM-2 could probably provide, in a cost-effective way, the large computational resources that our GA would need to find good solutions, and that the development process would be straightforward.

However, our GA does not fit the machine architecture as well as it might:

- The time required to compute the evaluation function for one generation of the GA—a process that requires no communication—far exceeds the communication overhead that is incurred when solutions are combined through crossover. Because much of the dollar cost of a parallel computer like the CM-2 is invested in the interprocessor communication network, it would appear that we are not making cost-effective use of the architecture.
- A population size of 4096 (one candidate solution per processor) is required to make full use of the CM-2, but it is probably too large for our application. Typical GAs use population sizes of 200 or fewer, and the increased size of our population is at the expense of additional generations.
- For many other applications, a steady-state GA outperforms a generational-replacement GA on a serial machine (Davis, 1991). However, a steady-state GA cannot be parallelized in a straighforward manner on a SIMD machine.

[2] Both behaviors are cyclic. Our representation fosters but is not restricted to cyclic behavior.

In our particular application the evaluation of candidate solutions turned out to be difficult to implement efficiently on a SIMD machine. To evaluate the "brain configuration" of a linkage requires running a physical simulation. Unfortunately, some requirements of the simulation are incompatible with the CM-2 architecture:

- The robust simulation of an articulated figure requires the checking of many special cases. This results in a great amount of conditionally executed code, and therefore wasted cycles on a SIMD machine.
- The key to efficient physical simulations of the kind we consider is the use of variable-length time steps. The SIMD architecture essentially mandates a uniform time step for all processors, resulting in slower and less robust simulations.

Our tentative conclusion is that, on balance, the CM-2 is not the ideal machine for our current application, even though we are delighted with the results that we have obtained. We are optimistic that more current parallel machines (for example, the CM-5, which has a more coarse-grained, MIMD architecture) may well address some of the issues that we raise above, and that our algorithm may benefit from architectural advances by computer manufacturers.

References

Badler, N. I., Barsky, B. A., and Zeltzer, D., editors (1991). *Making Them Move: Mechanics, Control, and Animation of Articulated Figures.* Morgan Kaufmann, San Mateo, California.

Brotman, L. S. and Netravali, A. N. (1988). Motion interpolation by optimal control. *Computer Graphics*, 22(4):309–315.

Davis, L. (1991). *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, San Francisco.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems.* University of Michigan Press.

Hopcroft, J., Joseph, D., and Whitesides, S. (1984). Movement problems for 2-dimensional linkages. *SIAM Journal of Computing*, 13(3):610–629.

Ngo, J. T. and Marks, J. (1992). Physically realistic trajectory planning in animation: A stimulus-response approach. Technical Report 21-92, Center for Research in Computing Technology, Harvard University, Cambridge, MA 02138.

Skinner, B. F. (1938). *The Behavior of Organisms; An Experimental Analysis.* The Century Psychology Series. D. Appleton-Century, New York, London.

Witkin, A. and Kass, M. (1988). Spacetime constraints. *Computer Graphics*, 22(4):159–168.