Limitations of parallel global optimization for large-scale human movement problems

Byung-Il Koh, Jeffrey A. Reinbolt, Alan D. George, Raphael T. Haftka, Benjamin J. Fregly

1. Introduction

Optimization methods are frequently used to develop human movement simulations that reproduce experimentally measured motions or predict new motions for which experimental data are not available [1–19]. These simulations typically utilize complex musculoskeletal computer models possessing multiple degrees of freedom (DOFs). Since 10 or more design variables are often used to parameterize the control of each DOF [1,3,17,20], the resulting movement optimization problems can possess hundreds of design variables, increasing the likelihood of encountering local minima. These large-scale problems can also have a high computational cost due to iterative evaluation of the cost function and constraints. Though ineffective against local minima, gradient-based optimization algorithms have commonly been used to solve large-scale problems, primarily because of their rapid convergence properties. However, even with these algorithms, the complexities of present-day biomechanical models can necessitate thousands of function evaluations to achieve convergence [9,21]. To increase throughput, researchers have developed and evaluated several parallel algorithms for biomechanical optimization problems. Gradient-based optimization algorithms were the first to be parallelized for human movement problems [4,5,22], and more recently, global optimization algorithms such as genetic [13], simulated annealing [14], and particle swarm [23–26] have been parallelized as well. Global algorithms are advantageous since they decrease the risk of being trapped in a local minimum, but the price is significantly higher computational cost compared to a gradient-based algorithm. To date, only gradient-based algorithms have been used for large-scale (>100 design variables) human movement problems (Table 1, top half) such as the three-dimensional jumping and gait problems with over 800 design variables solved by Anderson and Pandy [4,5]. Since global algorithms have been used successfully to solve small- to medium-scale human move-
Table 1  

<table>
<thead>
<tr>
<th>Study</th>
<th>Optimization algorithm</th>
<th>Model dimensionality</th>
<th>Simulated movement</th>
<th>Parallel implementation</th>
<th>Design variables</th>
<th>Body segments</th>
<th>Degrees of freedom</th>
<th>Observation movement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fregly et al. (2007) [7]</td>
<td>NLS</td>
<td>3D</td>
<td>Walking</td>
<td>No</td>
<td>27</td>
<td>12</td>
<td>580</td>
<td>Walking</td>
</tr>
<tr>
<td>Neptune (1999) [9]</td>
<td>SA</td>
<td>2D</td>
<td>Pedaling</td>
<td>No</td>
<td>3</td>
<td>7</td>
<td>27</td>
<td>Pedaling</td>
</tr>
<tr>
<td>Neptune et al. (2000) [10]</td>
<td>SA</td>
<td>3D</td>
<td>Running</td>
<td>No</td>
<td>9</td>
<td>20</td>
<td>43</td>
<td>Running</td>
</tr>
<tr>
<td>Ogihara and Yamazaki (2001)</td>
<td>GA</td>
<td>2D</td>
<td>Walking</td>
<td>No</td>
<td>7</td>
<td>7</td>
<td>93</td>
<td>Walking</td>
</tr>
<tr>
<td>McLean et al. (2003) [12]</td>
<td>SA</td>
<td>3D</td>
<td>Side-stepping</td>
<td>No</td>
<td>12</td>
<td>5</td>
<td>61</td>
<td>Side-stepping</td>
</tr>
<tr>
<td>van Soest and Casius (2003)</td>
<td>GA</td>
<td>2D</td>
<td>Pedaling</td>
<td>Yes</td>
<td>5</td>
<td>5</td>
<td>34</td>
<td>Pedaling</td>
</tr>
<tr>
<td>Higginson et al. (2005) [14]</td>
<td>SA</td>
<td>2D</td>
<td>Pedaling</td>
<td>Yes</td>
<td>3</td>
<td>7</td>
<td>27</td>
<td>Pedaling</td>
</tr>
<tr>
<td>Yang et al. (2007) [16]</td>
<td>SA</td>
<td>2D</td>
<td>Walking, Standing</td>
<td>No</td>
<td>9</td>
<td>7</td>
<td>88</td>
<td>Walking, Standing</td>
</tr>
</tbody>
</table>

2. Methods

A parallel particle swarm algorithm was chosen as the global optimization algorithm for attempting two large-scale human movement problems. This algorithm has been tested extensively and found to perform well on a number of small- and medium-scale benchmark problems [26]. For comparison purposes, both large-scale human movement problems were also solved using the Levenberg–Marquardt nonlinear least squares (NLS) algorithm [27] available in Matlab (The Mathworks, Natick, MA). This gradient-based algorithm was selected since both optimization problems were formulated as tracking problems that minimized sums of squares of errors, the situation for which nonlinear least squares algorithms are designed. In general terms, gradient-based algorithms calculate a search direction in multi-dimensional design space, step in that direction until a minimum is found, and then iterate the entire process until no further improvement can be made. The following sections describe the parallel particle swarm algorithm and the two large-scale human movement problems used for evaluation.

2.1. Parallel particle swarm optimization algorithm

Particle swarm global optimization is a class of derivative-free, population-based computational methods introduced by Kennedy and Eberhart [28]. In the original PSO algorithm, particles (design points) are distributed throughout the design space and their positions and velocities are modified based on knowledge of the best solution found thus far by each particle in the “swarm.” Attraction toward the best-found solution occurs stochastically and uses dynamically adjusted particle velocities. Particle pseudo-velocities (Eq. (1)) and positions (Eq. (2)) are updated as shown below:

\[
\begin{align*}
\dot{v}_{i+1}^x &= w \dot{v}_{i}^x + c_1 r_1 (p_k^x - x_i^x) + c_2 r_2 (p_{gb}^x - x_i^x) \\
\dot{x}_{i+1}^x &= x_i^x + \dot{v}_{i+1}^x
\end{align*}
\]  

where \(x_i^x\) represents the current position of particle \(i\) in design space and subscript \(k\) indicates a (unit) pseudo-time increment. The point \(p_k^x\) is the best-found position of particle \(i\) up to time step \(k\) and represents the cognitive contribution to the search velocity \(\dot{v}_{i+1}^x\). The point \(p_{gb}^x\) is the global best-found position among all particles in the swarm up to time step \(k\) and forms the social contribution to the velocity vector. Random numbers \(r_1\) and \(r_2\) are uniformly distributed in the interval \([0, 1]\). Parameters \(c_1\) and \(c_2\) are the cognitive...
and social scaling parameters, respectively, whose values are generally chosen such that their sum equals 4 [28]. Choosing $c_1 = c_2 = 2$ causes the particles to overshoot the attraction points $p^*_k$ and $p^*_k$ half the time, thereby maintaining separation in the group while allowing a greater area to be searched than if the particles did not overshoot. The value of the variable $w$, set to 1 at initialization, is reduced dynamically based on the cost function improvement rate, thereby reducing the search area gradually (see Ref. [25] for further details).

Two parallel PSO algorithms have been reported in the literature. One algorithm is called parallel synchronous PSO (PSPSO) [25], where the master processor updates particle positions and velocities only after all function evaluations have been completed by all slave processors. The other algorithm is called parallel asynchronous PSO (PAPSO) [26,29], where the master processor updates particle positions and velocities continuously as function evaluations are completed by the slave processors. Since PAPSO tends to find a slightly lower cost function value than does PSPSO for the same number of function evaluations [26], a parallel PAPSO algorithm was used in the present study.

Ten PAPSO runs were performed for both large-scale movement optimization problems. Each run was performed on a homogeneous Linux cluster of 20 identical PCs, utilized 20 particles along with other standard algorithm parameter values (see Ref. [24] for details), and iterated for 1 million function evaluations to ensure convergence. One of the particles was always assigned to match the nominal experimental gait motion, while the other 19 particles were given random initial guesses within the initial bounds of the design space. During each optimization, particles were allowed to “stray” beyond the initial bounds. Similar to previous optimization algorithm evaluations studies [13,24], no tuning of algorithm parameter values was performed for any of the optimizations. Execution time for each PAPSO optimization was approximately 2.3 h.

2.2. Large-scale human movement problems

Previously reported experimental gait and isolated joint motion data collected from a single patient with knee OA (male, age 37 years, height 170 cm, mass 69 kg, alignment 5° varus) were used as the initial guess for the two benchmark human movement optimization problems [8]. Institutional review board approval and informed consent were obtained. The gait and isolated joint motion data were used to calibrate the joint and inertial parameter values in a dynamic, full-body, three-dimensional (3D) gait model. The model possessed 27 DOFs composed of gimbal (3 DOFs), universal (2 DOFs), and pin (1 DOF) joints, where each DOF was chosen as a generalized coordinate in the model (Fig. 1). The position and orientation of the pelvis in the laboratory coordinate system were defined by three translational and three rotational DOFs, resulting in six pelvis residual loads (three forces and torques) during inverse dynamic analyses. For the lower body, each hip was modeled as a gimbal joint, each knee as a pin joint, and each ankle as two non-intersecting and non-orthogonal pin joints. For the upper body, the back was modeled as a gimbal joint, each shoulder as a universal joint, and each elbow as a pin joint. Ground reaction forces and torques applied to the feet were treated as unknowns to be determined during periods when the foot was known to be in contact with the ground [30]. Details on the experimental data collection and model calibration process can be found in Ref. [8].

The calibrated model and nominal gait data were used in two large-scale human movement optimization problems. Both problems involved predicting gait changes that reduce the left knee adduction torque (henceforth called the “toe out gait optimization”) [31], while the second problem predicted changes in gait motion that would decrease the left knee adduction torque without altering the toe out angle (henceforth called the “modified gait optimization”). The reality constraints minimized changes in trunk rotations, pelvis residual loads, and footpaths (with increased toe out angle where necessary) away from the nominal gait data.

The two gait prediction problems were formulated as inverse dynamic optimizations. Each optimization performed repeated inverse dynamics analyses, where each analysis used the patient-specific gait model to calculate joint torque and pelvis residual load outputs given the current guess for the joint angle and ground reaction inputs. The joint angle and ground reaction curves measured experimentally were parameterized as a function of time using a combination of polynomial and Fourier terms [32], thereby allowing analytic calculation of joint velocities and accelerations.

To match the experimental gait data accurately, a cubic polynomial with eight Fourier harmonics (i.e., 20 coefficients) was selected, producing root-mean-square (RMS) errors between experimental and fitted curves on the order of 0.3 mm, 0.2°, 4 N, and 0.7 Nm [7]. The design variables for the two optimization problems were defined as the coefficients of the parameterized curves. By changing the joint angle and ground reaction design variables, the optimizations were able to predict novel gait motions that satisfied the dynamics equations [6–8,30]. For both problems, the following
motion and ground reaction curves were allowed to vary: all pelvis translations, all back, pelvis, hip, knee, and ankle rotations, and all ground reaction forces and torques (33 curves in all for a total of 660 design variables). Shoulder and elbow rotations were prescribed to match the experimental data. Ground reaction forces and torques were set to zero for time frames when the foot was known to be off the floor [30].

Since the PAPSO and NLS algorithms can only solve unconstrained optimization problems, the constraints for both movement prediction problems were handled using a penalty method. The form of the augmented cost function is indicated in Eq. (3) below:

\[
\text{min} \sum_{j=1}^{nframes} \left\{ T_{\text{Addf}}^2 + \sum_{s=1}^{2} \left( \sum_{j=1}^{3} (\Delta T_{\text{Hip}})^2_{js} + (\Delta T_{\text{Knee}})^2_{js} \right) + \sum_{j=1}^{2} (\Delta T_{\text{Ankle}})^2_{js} + (\Delta q_{\text{CoP}})^2_{js} \right\} + \sum_{s=1}^{2} \left( \sum_{j=1}^{3} (\Delta q_{\text{Hip}})^2_{js} + (\Delta q_{\text{Knee}})^2_{js} + (\Delta q_{\text{Ankle}})^2_{js} \right) \]

\[
+ \sum_{j=1}^{6} (\Delta q_{\text{Pelvis}})^2_{js} + w_{\text{Trunk}} \sum_{j=1}^{3} (\Delta q_{\text{Trunk}})^2_{js} + w_{\text{Pelvis}} \sum_{j=1}^{6} (\Delta q_{\text{Pelvis}})^2_{js} + w_{\text{FootPath}} \sum_{s=1}^{2} \left( \sum_{j=1}^{6} (\Delta q_{\text{Foot}})^2_{js} \right) \}
\]

(3)

In this equation, \( nframes \) is the number of time frames used in the problem (=101), \( f \) is a specific time frame, \( j \) is a translational or rotational joint axis number, and \( s \) is a side. The weights \( w_{\text{Trunk}}, w_{\text{Pelvis}}, \) and \( w_{\text{FootPath}} \) (=10) for the reality constraints were initialized to 1 and increased by factors of 10 until RMS errors in trunk orientation, pelvis residual loads, and footpath were less than 5 mm, deg, N, or Nm. \( T_{\text{Add}} \) is the left knee adduction torque, which was removed for the toe out gait optimization and minimized for the modified gait optimization. Other constraints were changes in the following quantities away from their nominal experimental trajectories: hip flexion/extension, abduction/adduction, and inter/external rotation torque and angle (\( \Delta T_{\text{Hip}}, \Delta q_{\text{Hip}} \)), knee flexion/extension torque and angle (\( \Delta T_{\text{Knee}}, \Delta q_{\text{Knee}} \)), ankle flexion/extension and inversion/eversion torque and angle (\( \Delta T_{\text{Ankle}}, \Delta q_{\text{Ankle}} \)), anterior/posterior and medial/lateral center of pressure position (\( \Delta \text{CoP} \)), pelvis translations and rotations (\( \Delta q_{\text{Pelvis}} \)), trunk rotations (\( \Delta q_{\text{Trunk}} \)), pelvis residual forces and torques (\( \Delta T_{\text{Pelvis}} \)), and foot translations and rotations (\( \Delta q_{\text{Foot}} \)). Though foot translations and rotations and trunk rotations with respect to the laboratory coordinate system were not generalized coordinates in the model, these quantities were calculated from the current values of the generalized coordinates.

### Table 2

<table>
<thead>
<tr>
<th>Problem</th>
<th>Optimizer</th>
<th>Final cost function value</th>
<th>Adduction torque reduction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toe-out gait</td>
<td>PSO</td>
<td>1.71 × 10^3</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>NLS</td>
<td>6.67 × 10^4</td>
<td>2.9</td>
</tr>
<tr>
<td>Modified gait</td>
<td>PSO</td>
<td>7.55 × 10^6</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>NLS</td>
<td>2.36 × 10^4</td>
<td>32.4</td>
</tr>
</tbody>
</table>

3. Results

In contrast to the NLS algorithm, the PAPSO algorithm converged to an improved solution only for the toe out gait optimization problem. For both problems, the best final cost function value found by 10 PAPSO runs was larger than the final cost function value found by one NLS run (Table 2). With the exception of footpath rotation error for the PAPSO toe out gait optimization, all optimizations successfully satisfied the three reality constraints (trunk orientations, pelvis residual loads, and footpath translation/rotation), maintaining RMS errors to within 2 mm, 2°, 4 N, and 4 Nm.

For the toe out gait optimization, the PAPSO algorithm predicted little change in the first knee adduction torque peak and a 9.3% reduction in the second peak, while the NLS algorithm predicted little change in the first peak and a 32.2% reduction in the second one (Fig. 2). The PAPSO algorithm achieved only a 10° increase in toe out angle, whereas the NLS algorithm achieved the full 15° increase. The corresponding lateral shift in the CoP was larger for the NLS algorithm than for the PAPSO algorithm (Fig. 3). To achieve an increased foot progression angle, both algorithms predicted increased hip abduction, greatly increased hip external rotation, slightly decreased ankle flexion, and increased ankle eversion during stance phase.

For the modified gait optimization, PAPSO was unable to produce any adduction torque reductions, while NLS predicted reductions of 32.4% and 29.4% in the first and second peak, respectively (Fig. 4). These reductions were accompanied by a small lateral shift in the CoP (Fig. 5). The predicted kinematic changes were decreased pelvic obliquity, slightly increased hip, knee, and ankle flexion, and increased hip internal rotation, resulting in a medial shift of the left knee center.

![Fig. 2](image-url)  
Left knee adduction torque predictions for the toe out gait optimizations. Solid line is nominal experimental curve, dashed line is prediction generated by a single run with the nonlinear least squares algorithm (NLS), and dotted line is best prediction generated by 10 runs with the parallel asynchronous particle swarm algorithm (PSO).
4. Discussion

This study evaluated whether a global particle swarm optimization algorithm could solve two large-scale human movement problems possessing 660 design variables. Both problems utilized a three-dimensional, 27 DOF, full-body gait model whose joint and inertial parameters were calibrated to gait data collected from a single patient with knee OA. The cost function for both problems sought to reduce the peak knee adduction torque either by increasing the foot progression angle (i.e., the toe out gait optimization) or by minimizing the adduction torque directly without any changes in footpath (i.e., the modified gait optimization). Penalty terms were included to ensure that the predicted gait motions were physically realistic. The PAPSO algorithm achieved an improved solution only for the toe out gait optimization problem, whereas a gradient-based NLS algorithm achieved significant and realistic improvements for both problems. These findings suggest that global optimization algorithms proven to work well for small- to medium-scale problems may not work well for large-scale human movement problems. Researchers should not assume that an optimization solution is the global minimum simply because a global algorithm was used to generate it, and they should exercise caution when extrapolating the performance of global optimizers to problems involving hundreds of design variables.

There are at least three reasons why the PAPSO algorithm performed worse than did the NLS algorithm on these two benchmark problems. First, no tuning was performed of PAPSO algorithm parameters. It is possible that our PAPSO algorithm could have performed better had time-consuming algorithm parameter tuning been performed. Such tuning was not performed in the present study since few researchers take the time to do it (i.e., researchers want to find algorithms that work well “out of the box”), and since studies that evaluate optimization algorithm performance normally do not tune the algorithm parameters to the specific problems being investigated [13,24]. Second, the shape of the design space made it difficult to locate the global minimum without the use of gradient information. Though the design space was smooth for both sample problems, the minimum was located in a small deep hole in design space (Fig. 6), making it extremely difficult for a particle to migrate into this hole using only stochastic updates. Third, only a single set of penalty weights was used for both optimizations problems rather than ramping up the weights slowly over a sequence of optimization runs [33]. A sequence of optimizations with increasing penalty weights was not performed because of the excessive computation time required by this approach. Fur-
thermore, the selected weights posed no problem for the NLS algorithm, suggesting that the PSO algorithm may have convergence difficulties for problems involving penalty terms. Differences in optimization algorithm performance cannot be attributed to differences in problem formulation, since both algorithms used exactly the same formulations. In fact, the PSO algorithm possesses the advantage of insensitivity to design variable scaling, which gradient-based algorithms with finite-difference gradients do not possess [24].

Since the global optimization results reported in this study were based on a single optimization algorithm, it is not clear that similar poor performance would be observed for other global algorithms. However, this will likely be the case based on previous benchmark problems solved with a synchronous version of our PSO algorithm and several other global algorithms [24]. At a minimum, researchers who desire to use a global optimization algorithm on a large-scale human movement problem should compare their results with those from a gradient-based algorithm as a check.

The best result for both optimizations was consistent with recent experimental observations. The toe out gait optimization predicted that increasing the foot progression angle by 15° would reduce the second but not the first peak of knee adduction torque curve. When Guo et al. [34] increased the experimental toe out angle by 15°, they found little change in the first adduction torque peak and roughly a 40% reduction in the second one. When the subject in our study increased his toe out angle by approximately 15°, the experimental reduction in the second peak was approximately 30% with little change in the first one [8]. By comparison, our NLS toe out gait optimization predicted a 32.2% reduction in the second peak and only a slight change in the first one, with both peaks being within ±1 S.D. of the mean values measured from our subject over three trials. The modified gait optimization predicted that internally rotating the hips so as to medialize the knees would reduce both peaks of the knee adduction torque curve. When Barrios and Davis [35] recently trained a healthy subject to walk with the hips internally rotated and the knees medialized, they observed a 28% reduction in the first knee adduction torque peak. Our NLS modified gait optimization predicted a comparable reduction of 32.4%.

In summary, this study evaluated the ability of a global particle swarm optimization algorithm to solve large-scale human movement problems. Though the PSO algorithm used in our study has performed well on small- to medium-scale benchmark and biomechanical optimization problems [26], it did not perform well on the two large-scale human movement problems investigated in this study, with a gradient-based nonlinear least squares algorithm performing better on both problems. Significant algorithm parameter tuning or use of a global–local hybrid algorithm may be necessary for PSO and other global optimizers to solve large-scale human movement problems. Based on the results of this study, the PSO algorithm is not recommended for solving large-scale human movement optimization problems possessing constraints or competing terms in the cost function. Researchers should exercise caution when evaluating the results of large-scale human movement optimization problems solved using global algorithms.

Acknowledgments

This study was supported by NIH National Library of Medicine Grant R03LM007332, a Whitaker Foundation Biomedical Engineering Research Grant, and NIH National Center for Medical Rehabilitation Research Grant R21HD053490 to B.J.F.

Conflict of interest

None of the authors of this manuscript have any conflict of interest related to this work.

References


