Purpose: To investigate and evaluate fast two-point step-size gradient methods for IMRT/VMAT treatment plan optimization with nonlinear convex models.

Methods: Commonly applied optimization models for IMRT/VMAT planning are nonlinear and convex, including many quadratic physical models and general nonlinear biological models. The solutions are typically found through gradient-based algorithms. Existing methods, however, suffer from the low efficiency of the line search process. In this work, we investigate the use of two-point step-size gradient methods for solving general nonlinear convex models in IMRT/VMAT planning. With such algorithms, the line search step can be avoided or highly reduced, and significant speedup can be obtained. As a specified form of nonlinear convex models, the quadratic models are particularly investigated and the best form of the gradient method without line search is found.

Results: Five clinical prostate and five clinical head-and-neck cancer cases were tested for both IMRT and VMAT planning. For general nonlinear convex models, the Modified Two-Point Step-Size Gradient Method was found to have the best efficiency: for all test cases, approximately 5 times speedup was obtained with similar convergence properties compared to the traditional line search method. For quadratic models, the original Barzilai-Borwein method offered the best performance: for all test cases, approximately 10 times speedup was obtained with better convergence properties (i.e., better treatment quality). For instance, for typical prostate cancer cases with a penalty-based quadratic model, it takes ~4 seconds to generate optimized 9-beam fluence maps compared to ~40 seconds using traditional line search method on a PC.

Conclusions: This work provides a guideline to speedup IMRT/VMAT treatment plan optimization for general nonlinear convex models without loss of treatment quality. The current experiments were all run on CPU. Based on our previous experience, additional 10-30 times speedup can be expected with GPU-based implementation.