適応的近傍並列 SA における解通信の導入

伏見 俊彦

1　前回からの課題

- 提案手法に，解通信を導入し，性能向上の検討．

2　今月の研究内容

2.1　解通信の導入

連続最適化問題に SA を適用する場合は 複数ある SA のパラメータの中でも近傍が最も重要なパラメータである．近傍を並列化にすることにより，最適な近傍の値が分からなくても，対象問題に応じて自律的に最適な近傍で探索を行えるアルゴリズムを組み込んだものが適応的近傍並列 SA である．

今月は，適応的近傍並列 SA に解通信を導入することで，探索性能の向上の有無の検証を行った．解通信とは各クーリング周期ごとに同期をとる際に，各プロセスが持つエリート解の中で，最も優れたものを他のプロセスに分配し，分配した解から再び探索を行うことである．各プロセスはそれぞれ異なる近傍を持っているため，良好な解が様々な近傍を持つことになる．

3　実験結果

3.1　解探索性能

解通信を行う場合に，同期時に最も良好な解を分配する確率を分配率として，分配率を 0% ～ 100% で 5 パターンについて検証を行った．対象問題は Rastrigin 関数を用い，2 次元～10 次元についての結果を Fig. 1 に示す．縦軸はエネルギー，横軸は次元を示している．

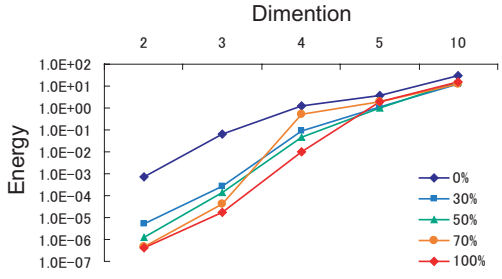

Fig. 1　Rastrigin 関数の結果

Fig. 1 より，低次元の問題では分配率を大きいものが良好な結果を示している．しかし，高次元の問題になるにつれて，分配率による差が見られなくなっている．

3.2　探索近傍幅

解通信を行わない場合と行う場合で探索近傍幅の収束の仕方に変化が見られた．探索開始時から終了時までの近傍幅の推移を Fig. 2，Fig. 3 に示す．縦軸は近傍，横軸はクーリング周期をそれぞれ示している．

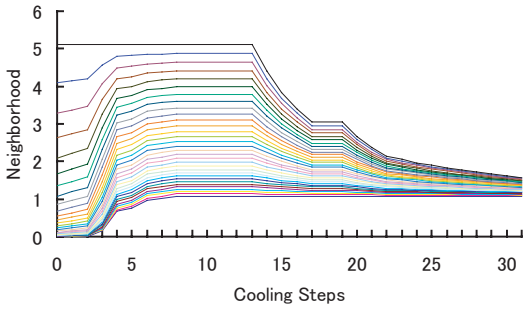

Fig. 2　解通信を行わない場合の近傍の推移

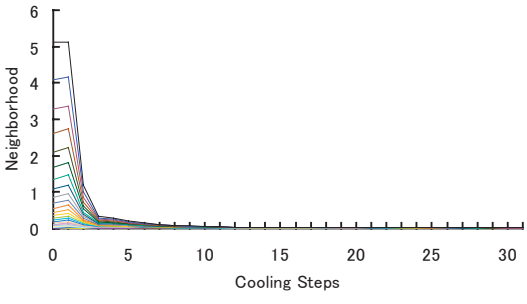

Fig. 3　解通信を行う場合の近傍の推移

解通信を行わない場合は近傍の収束が緩やかである．一方，解通信を行う場合は探索序盤で急速に収束が起こっている．この原因は，解通信を行わない場合は探索開始時では小さい近傍をもつプロセスは局所解に陥る．よって大きい近傍を持つプロセスが良好な解を得る．その結果小さい近傍が削除される．

解通信を行う場合は同期時に良好な解が他のプロセスに分配されるため，小さい近傍を持つプロセスが次の同期時に良好な解を得る．その結果大きい近傍が削除されると考えている．

4　今後の課題

今回の結果より，解探索性能だけに注目すると，解通信を行う方が良好な結果を示している．しかし，近傍の推移を見ると探索過程で近傍が急速に小さくなり過ぎているため，局所解への収束が考えられる．よって，今後は GA を用いて適応的に近傍を決定する予定である．