The testing of the 4 minimization methods on 10 greedy algorithm generated permutations. The functions can be found in TestingFunction.nb.

- First generate greedy algorithm-generated permutations. Note, for a better optimization, the function JonesResult should be used. Using JonesOptimize is essentially useless.

```
In[∘]:= checkGreedyDataSet =
          Table[bestGreedy[sampleKnot, {factor → JonesOptimize, tries → 1000}], {i, 10}];

In[∘]:= JonesResult[#] &/@ checkGreedyDataSet

Out[∘]= {211 823 004, 709 489 940, 524 374 740, 43 315 034, 857 994 188,
         61 321 364, 40 848 768, 329 420 220, 115 968 744, 5 926 675 980}
```

- Local Minimization Testing - Each process is repeated twice, in the second run the first number in each bracket shows the time it takes to perform the Local Minimization Process

```
In[∘]:= JonesResult @ localOptimization[PD @@ #,
            {startPt → convertToList, invariantCriteria → JonesOptimize, optSize → 3}] &/@
         checkGreedyDataSet

Out[∘]= {211 493 476, 686 421 502, 459 790 890, 39 085 516, 857 738 158,
         48 128 756, 25 126 838, 218 529 492, 114 863 208, 1 552 144 186}

In[∘]:= (Timing @ (JonesResult @ localOptimization[PD @@ #,
              {startPt → convertToList, invariantCriteria → JonesOptimize, optSize → 3}])) &/@
         checkGreedyDataSet

Out[∘]= {{0.046875, 211 493 476}, {0.015625, 686 421 502}, {0.03125, 459 790 890},
         {0.03125, 39 085 516}, {0.015625, 857 738 158}, {0.015625, 48 128 756}, {0.03125, 25 126 838},
         {0.03125, 218 529 492}, {0.015625, 114 863 208}, {0.015625, 1 552 144 186}}

In[∘]:= JonesResult @ localOptimization[PD @@ #,
            {startPt → convertToList, invariantCriteria → JonesOptimize, optSize → 4}] &/@
         checkGreedyDataSet

Out[∘]= {210 420 430, 570 133 408, 455 064 552, 32 758 806, 839 266 670,
         44 081 834, 24 927 726, 208 265 470, 114 829 460, 1 552 144 186}

In[∘]:= (Timing @ (JonesResult @ localOptimization[PD @@ #,
              {startPt → convertToList, invariantCriteria → JonesOptimize, optSize → 4}])) &/@
         checkGreedyDataSet

Out[∘]= {{0.09375, 210 420 430}, {0.09375, 570 133 408}, {0.09375, 455 064 552},
         {0.09375, 32 758 806}, {0.09375, 839 266 670}, {0.09375, 44 081 834}, {0.09375, 24 927 726},
         {0.09375, 208 265 470}, {0.09375, 114 829 460}, {0.09375, 1 552 144 186}}

In[∘]:= JonesResult @ localOptimization[PD @@ #,
            {startPt → convertToList, invariantCriteria → JonesOptimize, optSize → 5}] &/@
         checkGreedyDataSet

Out[∘]= {133 535 600, 547 921 376, 453 987 716, 32 758 806,
         286 180 940, 43 727 228, 24 681 530, 163 124 956, 114 544 318, 546 343 252}
```

```
In[∘]:= (Timing @ (JonesResult @ localOptimization[PD @@ #,
                {startPt → convertToList, invariantCriteria → JonesOptimize, optSize → 5}])) &/@
         checkGreedyDataSet

Out[∘]= {{0.546875, 133 535 600}, {0.53125, 547 921 376}, {0.5625, 453 987 716},
         {0.515625, 32 758 806}, {0.546875, 286 180 940}, {0.53125, 43 727 228}, {0.5, 24 681 530},
         {0.53125, 163 124 956}, {0.53125, 114 544 318}, {0.546875, 546 343 252}}

In[∘]:= JonesResult @ localOptimization[PD @@ #,
            {startPt → convertToList, invariantCriteria → JonesOptimize, optSize → 6}] &/@
         checkGreedyDataSet

Out[∘]= {133 239 490, 540 736 600, 373 446 554, 31 318 748,
         253 966 092, 43 727 228, 19 673 062, 156 781 900, 107 959 454, 217 298 546}

In[∘]:= (Timing @ (JonesResult @ localOptimization[PD @@ #,
                {startPt → convertToList, invariantCriteria → JonesOptimize, optSize → 6}])) &/@
         checkGreedyDataSet

Out[∘]= {{3.82813, 133 239 490}, {3.85938, 540 736 600}, {3.92188, 373 446 554},
         {3.73438, 31 318 748}, {3.84375, 253 966 092}, {3.79688, 43 727 228},
         {3.71875, 19 673 062}, {3.8125, 156 781 900}, {3.75, 107 959 454}, {3.85938, 217 298 546}}

In[∘]:= JonesResult @ localOptimization[PD @@ #,
            {startPt → convertToList, invariantCriteria → JonesOptimize, optSize → 7}] &/@
         checkGreedyDataSet

Out[∘]= {131 149 116, 457 682 970, 365 453 426, 31 230 878,
         246 388 764, 35 915 400, 18 834 582, 150 531 016, 51 734 732, 206 062 106}

In[∘]:= (Timing @ (JonesResult @ localOptimization[PD @@ #,
                {startPt → convertToList, invariantCriteria → JonesOptimize, optSize → 7}])) &/@
         checkGreedyDataSet

Out[∘]= {{30.0156, 131 149 116}, {30.0469, 457 682 970}, {30.4063, 365 453 426},
         {29.3125, 31 230 878}, {30.6719, 246 388 764}, {30.3906, 35 915 400}, {29.8438, 18 834 582},
         {30.6719, 150 531 016}, {29.6406, 51 734 732}, {30.2656, 206 062 106}}

In[∘]:= JonesResult @ localOptimization[PD @@ #,
            {startPt → convertToList, invariantCriteria → JonesOptimize, optSize → 8}] &/@
         checkGreedyDataSet

Out[∘]= {127 594 136, 331 387 766, 360 210 108, 15 594 922,
         246 185 956, 35 877 228, 18 813 524, 64 197 602, 51 734 732, 205 491 540}
```

- Partial Local Minimization - because this can have fluctuation, we will proceed

```
In[∘]:= N@ Mean[Table[JonesResult @ partialLO[PD @@ #,
                {startPt → convertToList, invariantCriteria → PartialJonesOptimize, optSize → 3,
                 partialPortion → 0.1, rounds → 10}], {i, 10}]] &/@ checkGreedyDataSet
```

Out[∘]= $\{2.05415\times10^{8}, 6.06227\times10^{8}, 4.48885\times10^{8}, 3.5831\times10^{7}, 7.95473\times10^{8}, 4.59562\times10^{7}, 2.48145\times10^{7}, 2.04856\times10^{8}, 1.11117\times10^{8}, 1.99314\times10^{9}\}$

```
In[]:= N@ Mean[Table[JonesResult @ partialLO[PD @@ #,
               {startPt → convertToList, invariantCriteria → PartialJonesOptimize, optSize → 4,
                partialPortion → 0.1, rounds → 10}], {i, 10}]] & /@ checkGreedyDataSet

Out[]= {1.30769×10^8, 5.2538×10^8, 4.26672×10^8, 3.17582×10^7, 5.4624×10^8,
        4.70929×10^7, 1.95129×10^7, 1.23973×10^8, 1.10783×10^8, 2.15756×10^8}

In[]:= N@ Mean[Table[JonesResult @ partialLO[PD @@ #,
               {startPt → convertToList, invariantCriteria → PartialJonesOptimize, optSize → 5,
                partialPortion → 0.1, rounds → 10}], {i, 10}]] & /@ checkGreedyDataSet

Out[]= {1.27226×10^8, 4.89383×10^8, 2.66183×10^8, 3.39758×10^7, 2.34801×10^8,
        4.24019×10^7, 1.92237×10^7, 6.83813×10^7, 1.0743×10^8, 2.0624×10^8}

In[]:= N@ Mean[Table[JonesResult @ partialLO[PD @@ #,
               {startPt → convertToList, invariantCriteria → PartialJonesOptimize, optSize → 6,
                partialPortion → 0.1, rounds → 10}], {i, 10}]] & /@ checkGreedyDataSet

Out[]= {1.25471×10^8, 3.84963×10^8, 1.60653×10^8, 2.07719×10^7, 2.37004×10^8,
        6.41246×10^7, 1.86877×10^7, 3.38396×10^7, 4.90462×10^7, 1.89889×10^8}

In[]:= N@ Mean[Table[JonesResult @ partialLO[PD @@ #,
               {startPt → convertToList, invariantCriteria → PartialJonesOptimize, optSize → 7,
                partialPortion → 0.1, rounds → 10}], {i, 10}]] & /@ checkGreedyDataSet

Out[]= {1.25108×10^8, 4.2805×10^8, 1.12465×10^8, 5.01152×10^6, 2.10466×10^8,
        4.16705×10^7, 1.88315×10^7, 1.69297×10^7, 4.05109×10^7, 9.29048×10^7}
```

- Reverse Local Minimization

```
storedGreedy = checkGreedyDataSet;

In[]:= JonesResult @ reverseLocalOptimization[PD @@ #,
           {startPt → convertToList, invariantCriteria → JonesOptimize, optSize → 3}] & /@
        storedGreedy

Out[]= {211 483 492, 694 719 990, 518 417 248, 39 085 516, 857 730 614,
        57 071 690, 37 687 550, 253 000 604, 114 863 208, 5 901 514 324}

In[]:= JonesResult @ reverseLocalOptimization[PD @@ #,
           {startPt → convertToList, invariantCriteria → JonesOptimize, optSize → 4}] & /@
        storedGreedy

Out[]= {210 314 862, 548 265 534, 512 638 088, 39 035 242, 789 348 984,
        54 319 512, 34 371 442, 247 034 662, 114 829 460, 5 869 022 396}

In[]:= JonesResult @ reverseLocalOptimization[PD @@ #,
           {startPt → convertToList, invariantCriteria → JonesOptimize, optSize → 5}] & /@
        storedGreedy

Out[]= {165 640 572, 529 596 744, 490 963 966, 35 710 814, 682 503 936,
        52 137 012, 30 648 602, 122 457 238, 114 827 326, 5 857 540 916}
```

```
In[∘]:= JonesResult @ reverseLocalOptimization[PD @@ #,
            {startPt → convertToList, invariantCriteria → JonesOptimize, optSize → 6}] & /@
         storedGreedy

Out[∘]= {165 624 702, 525 796 376, 395 689 986, 34 953 136,
         399 022 776, 49 963 942, 22 872 844, 63 478 422, 108 307 098, 5 735 062 568}

In[∘]:= JonesResult @ reverseLocalOptimization[PD @@ #,
            {startPt → convertToList, invariantCriteria → JonesOptimize, optSize → 7}] & /@
         storedGreedy

Out[∘]= {149 765 814, 654 215 542, 374 424 064, 28 472 360,
         328 611 900, 47 912 178, 22 045 186, 99 732 446, 100 398 626, 2 515 309 812}
```

- Reverse Partial Local Minimization

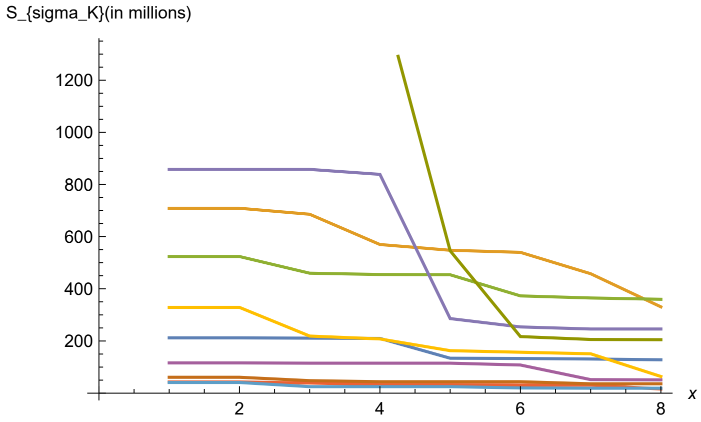
```
In[∘]:= convertToList[pd_PD] := List @@ pd;

In[∘]:= Protect[startPt, invariantCriteria, optSize, partialPortion, rounds];

In[∘]:= Options[backwardsPLO] :=
          {startPt → convertToList, invariantCriteria → PartialJonesOptimize,
           optSize → 3, partialPortion → 0.1, rounds → 3};

In[∘]:= backwardsPLO[pd_PD, opts : OptionsPattern[backwardsPLO]] :=
         Module[{s, k, sBest, optimizeList, optimizeResult, widths, width, round},
          s = OptionValue[startPt][pd];
          k = Length[s] - OptionValue[optSize] + 1;
          round = 1;
          sBest = s;
          While[round ≤ OptionValue[rounds],
           widths = FoldList[Complement [#1 ⋃ #2, #1 ⋂ #2] &, {}, List @@@ sBest];
           While[k ≥ 1,
            optimizeList = sBest⟦k ;; k + OptionValue[optSize] - 1⟧;
            width = widths⟦k⟧;
            optimizeResult = OptionValue[invariantCriteria][
              width, optimizeList, k, OptionValue[partialPortion]];
            sBest = Join[If[k == 1, {}, sBest⟦1 ;; k - 1⟧], optimizeResult,
              If[k == Length[s] - OptionValue[optSize] + 1,
               {}, sBest⟦k + OptionValue[optSize] ;; -1⟧]];
            k = k - 1;
           ];
           round = round + 1;
           k  = Length[s] - OptionValue[optSize] + 1;
          ];
          sBest]
```

```
In[•]:= N@ Mean[Table[JonesResult @ backwardsPLO[PD @@ #,
              {startPt → convertToList, invariantCriteria → PartialJonesOptimize, optSize → 3,
               partialPortion → 0.1, rounds → 10}], {i, 10}]] & /@ storedGreedy

Out[•]= {2.05701 × 10^8, 6.83088 × 10^8, 4.66842 × 10^8, 3.73616 × 10^7, 8.04095 × 10^8,
         4.68134 × 10^7, 2.54265 × 10^7, 1.96857 × 10^8, 1.12219 × 10^8, 4.86981 × 10^9}

In[•]:= N@ Mean[Table[JonesResult @ backwardsPLO[PD @@ #,
              {startPt → convertToList, invariantCriteria → PartialJonesOptimize, optSize → 4,
               partialPortion → 0.1, rounds → 10}], {i, 10}]] & /@ storedGreedy

Out[•]= {1.43842 × 10^8, 4.84177 × 10^8, 4.19224 × 10^8, 3.28005 × 10^7, 6.17519 × 10^8,
         4.24898 × 10^7, 1.95728 × 10^7, 1.26338 × 10^8, 1.1069 × 10^8, 6.17886 × 10^8}

In[•]:= N@ Mean[Table[JonesResult @ backwardsPLO[PD @@ #,
              {startPt → convertToList, invariantCriteria → PartialJonesOptimize, optSize → 5,
               partialPortion → 0.1, rounds → 10}], {i, 10}]] & /@ storedGreedy

Out[•]= {1.26806 × 10^8, 4.42854 × 10^8, 3.5545 × 10^8, 3.1153 × 10^7, 3.63733 × 10^8,
         3.35303 × 10^7, 1.88712 × 10^7, 7.50507 × 10^7, 1.05494 × 10^8, 2.09927 × 10^8}

In[•]:= N@ Mean[Table[JonesResult @ backwardsPLO[PD @@ #,
              {startPt → convertToList, invariantCriteria → PartialJonesOptimize, optSize → 6,
               partialPortion → 0.1, rounds → 10}], {i, 10}]] & /@ storedGreedy

Out[•]= {1.2052 × 10^8, 4.55325 × 10^8, 3.30374 × 10^8, 2.88247 × 10^7, 1.03258 × 10^8,
         3.0759 × 10^7, 1.8595 × 10^7, 3.81274 × 10^7, 4.59213 × 10^7, 1.75804 × 10^8}

In[•]:= N@ Mean[Table[JonesResult @ backwardsPLO[PD @@ #,
              {startPt → convertToList, invariantCriteria → PartialJonesOptimize, optSize → 7,
               partialPortion → 0.1, rounds → 10}], {i, 10}]] & /@ storedGreedy

Out[•]= {1.18742 × 10^8, 3.1701 × 10^8, 3.27035 × 10^8, 6.9134 × 10^6, 9.38896 × 10^7,
         3.03899 × 10^7, 1.8598 × 10^7, 5.04479 × 10^7, 4.31002 × 10^7, 2.47439 × 10^7}
```

- Plots Comparing the 4 Methods
- Plot 1: Effect of LM

```
In[∘]:= ListLinePlot[{{212, 212, 211, 210, 134, 133, 131,  128},
          {709, 709, 686, 570, 548, 540, 458, 331},
          {524, 524, 460, 455, 454, 373, 365, 360},
          {43, 43, 39, 33, 33, 31, 31, 16},
          {858, 858, 858, 839, 286, 254, 246, 246},
          {61, 61, 48, 44, 44, 44, 36, 36},
          {41, 41, 25, 25, 25, 20, 19, 19},
          {329, 329, 219, 208, 163, 157, 151, 64},
          {116, 116, 115, 115, 115, 108, 52, 51},
          {5927, 5927, 1552, 1552, 546, 217, 206, 205}},
        {AxesLabel → {x, "S_{sigma_K}(in millions)"}}]
```

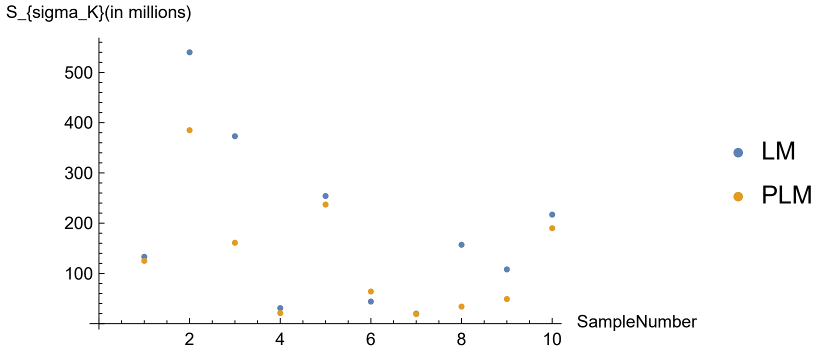
Out[∘]=



- Plot 2: LM vs PLM

```
In[∘]:= ListPlot[{{133, 540, 373, 31, 254, 44, 20, 157, 108, 217},
          {125, 385, 161, 21, 237, 64, 19, 34, 49, 190}},
        {AxesLabel → {"SampleNumber", "S_{sigma_K}(in millions)"},
         PlotLegends → {"LM", "PLM"}}]
```

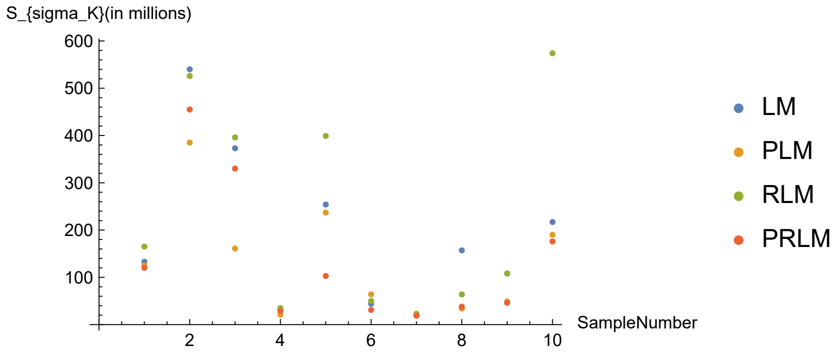
Out[∘]=



- Plot 3: Comparison of the 3 Methods

```
In[∘]:= ListPlot[{{133, 540, 373, 31, 254, 44, 20, 157, 108, 217},
          {125, 385, 161, 21, 237, 64, 19, 34, 49, 190},
          {165, 526, 396, 35, 399, 50, 23, 64, 108, 574},
          {120, 455, 330, 29, 103, 31, 19, 38, 46, 176}},
         {AxesLabel → {"SampleNumber", "S_{sigma_K}(in millions)"},
          PlotLegends → {"LM", "PLM", "RLM", "PRLM"}}]
```

Out[∘]=


S_{sigma_K}(in millions)
SampleNumber
● LM
● PLM
● RLM
● PRLM