Problema comis-voiajorului

1.Descreierea problemei

Problema comis-voiajorului este o problemă clasică și cea mai frecvent analizată problemă de optimizare combinatorică, cu o gamă largă de metodologii de rezolvare. Avem o listă de orașe și distanțele dintre acele orașe. O persoană are de vizitat acea listă de orașe, începând și sfârșind cu primul oraș din lista dată. Se cere să se găsească turul în care fiecare oraș este vizitat o singură dată și distanța parcursă este minimă.

2.Rezolvare prin algoritm genetic

[START] Generăm populația de indivizi, (indivizii sunt drumurile care satisfac condițiile menționate în descrierea problemei, iar populația este o colecție de drumuri), această primă populație fiind generația 1.

[FITNESS] Prin intermediul funcției de fitness, evaluăm fiecare individ din populație (funcția fitness verifică lungimea drumului).

[ELITE] Din populația curentă, păstrăm individul pentru care s-au obținut cele mai bune rezultate în funcția de fitness (drumul cu lungimea cea mai mică).

[NP] Creăm o populație nouă prin repetarea următoarelor etape, până la obținerea unei noi populații complete. Numărul generației va crește cu 1 și noua populație va înlocui vechea populație.

[SELECTION] Selectăm 2 părinți din generația anterioară

[CROSSOVER] Folosind operatorul de încrucișare, încrucișăm cei 2 indivizi aleși ca și părinți ca să obținem un nou individ (selectăm la întâmplare o secțiune a primului părinte ca bază a noului individ și completăm cu elementele celui de al doilea părinte).

[MUTATION] În funcție de probabilitatea de mutație, există posibilitatea ca noul individ să sufere mutații (două valori luate la întâmplare vor fi interschimbate). Indiferent dacă s-a produs mutația sau nu, adăugăm noul individ în populație.

[TEST&LOOP] Verificăm numărul de generații, dacă am ajuns la numărul maxim de generații, algoritmul se încheie și returnează cea mai bună soluție din populația curentă. Altfel, repetăm pașii de la [FITNESS] până la [TEST&LOOP].

3.Rulare în HeuristicLab

**3.1 Rulare cu parametrii de bază**

Elites: 1

MaximumGenerations: 1000

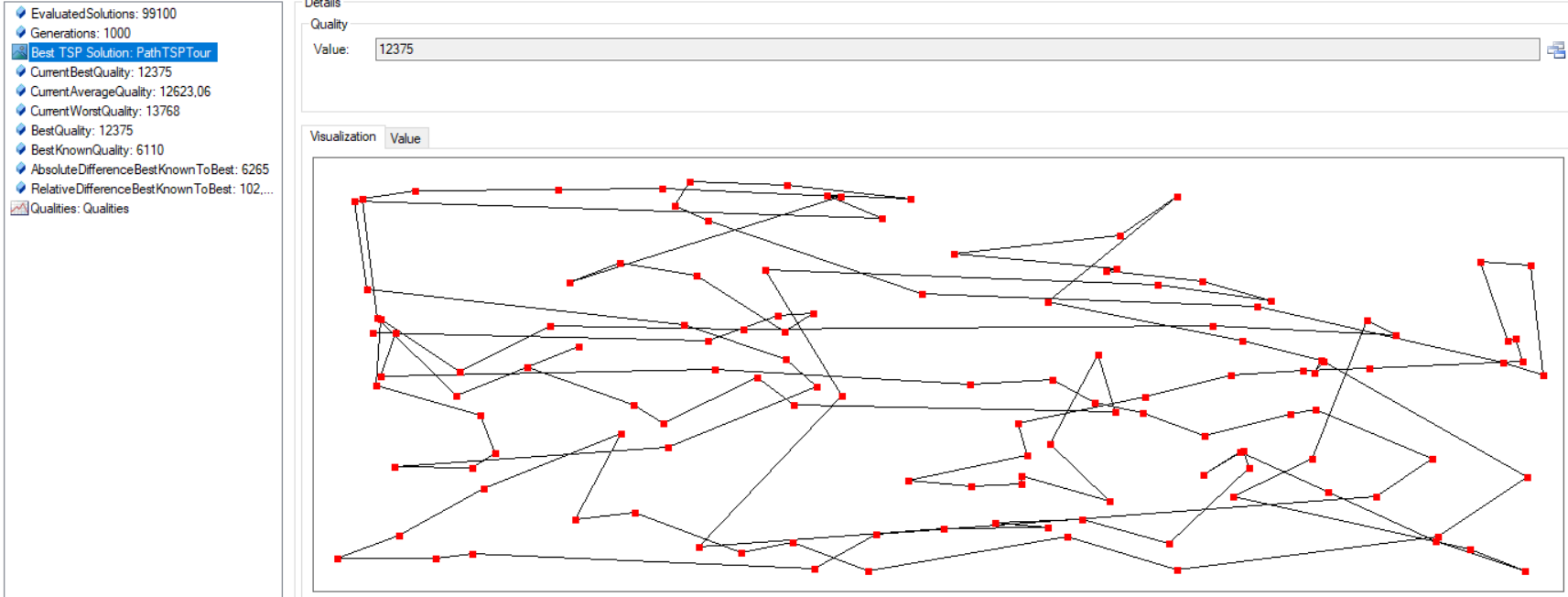
MutationProbability: 5%

PopulationSize: 100

Crossover: OrderCrossover2

Mutator: InversionManipulator

Selector: ProportionalSelector



**3.2 Rulare cu generație maximă 3000**

Elites: 1

MaximumGenerations: 3000

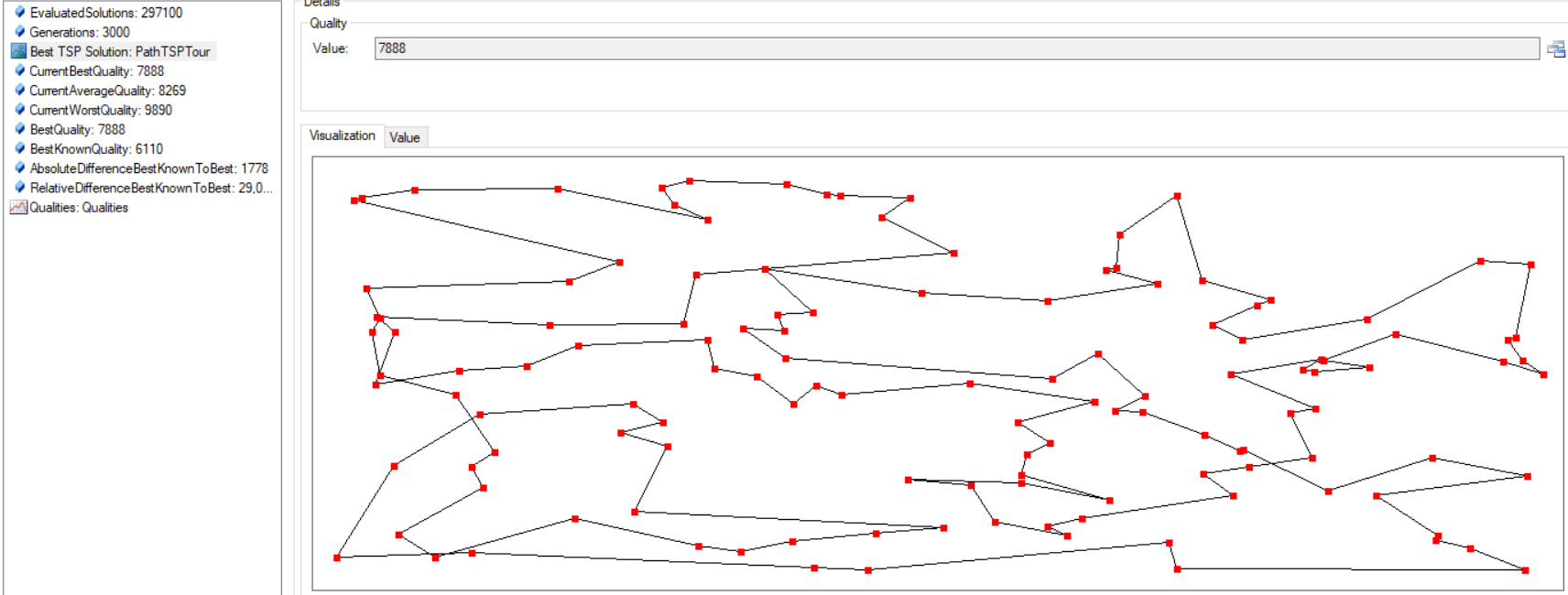
MutationProbability: 5%

PopulationSize: 100

Crossover: OrderCrossover2

Mutator: InversionManipulator

Selector: ProportionalSelector



**3.3 Rulare cu generație maximă 5000**

Elites: 1

MaximumGenerations: 5000

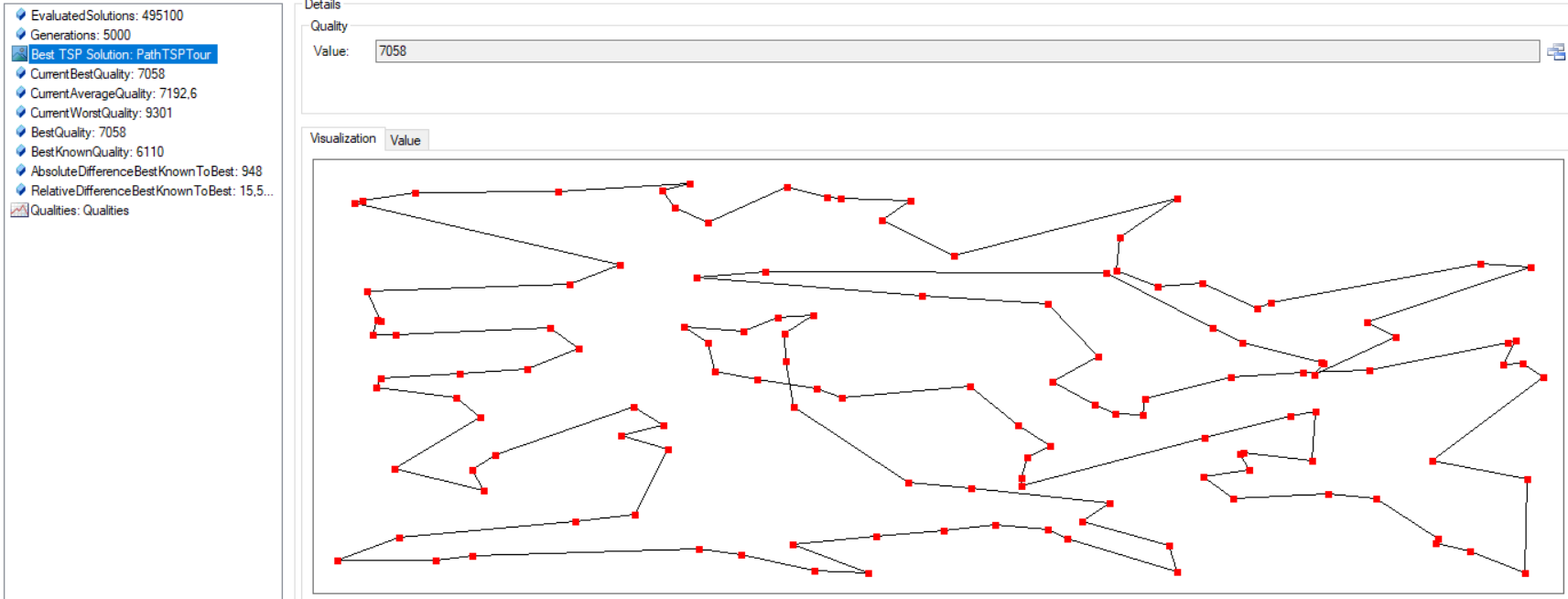
MutationProbability: 5%

PopulationSize: 100

Crossover: OrderCrossover2

Mutator: InversionManipulator

Selector: ProportionalSelector



**3.4 Rulare cu probabilitate de mutație 2%**

Elites: 1

MaximumGenerations: 1000

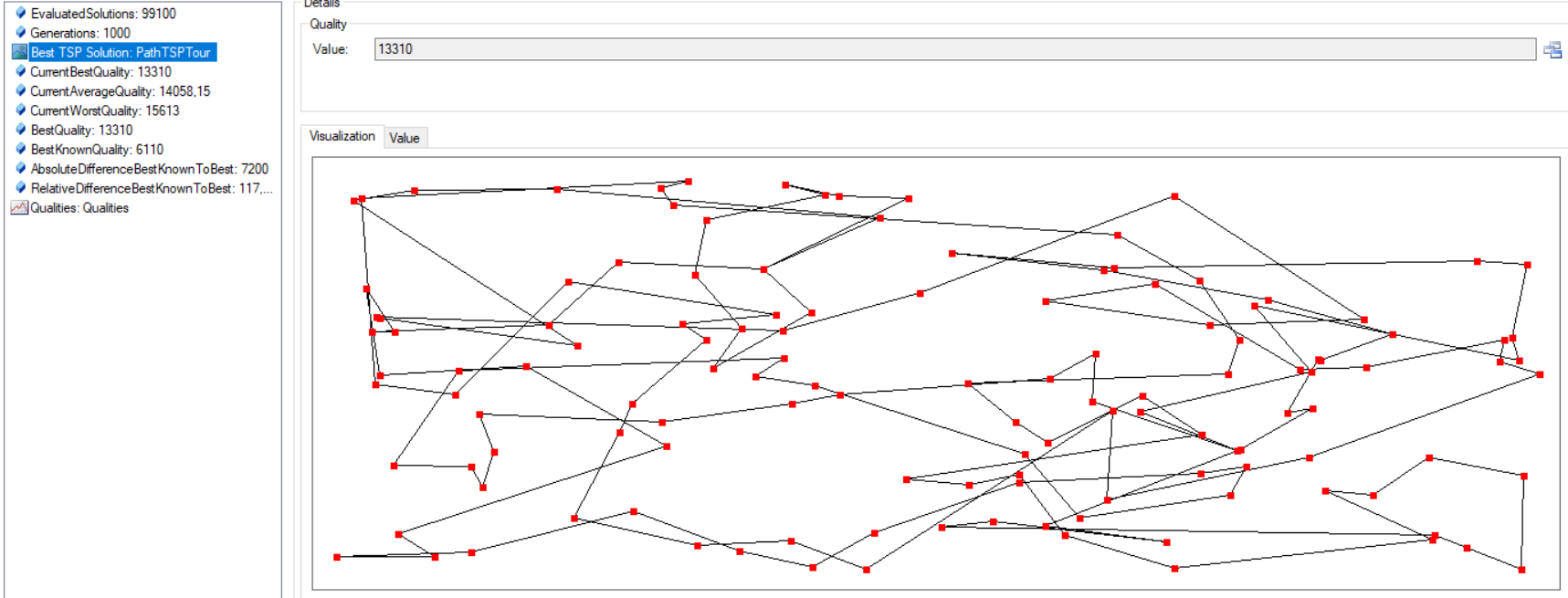
MutationProbability: 2%

PopulationSize: 100

Crossover: OrderCrossover2

Mutator: InversionManipulator

Selector: ProportionalSelector



**3.5 Rulare cu probabilitate de mutație 15%**

Elites: 1

MaximumGenerations: 1000

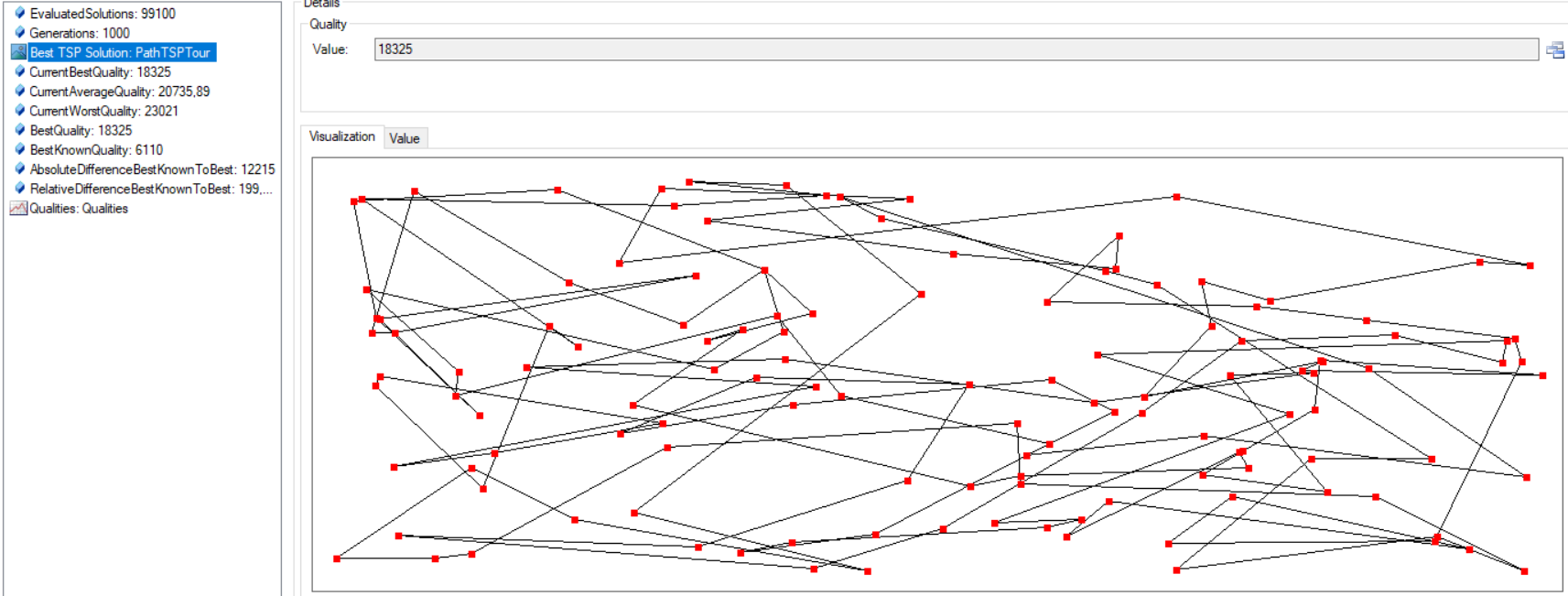
MutationProbability: 15%

PopulationSize: 100

Crossover: OrderCrossover2

Mutator: InversionManipulator

Selector: ProportionalSelector



**3.6 Rulare cu probabilitate de mutație 50%**

Elites: 1

MaximumGenerations: 1000

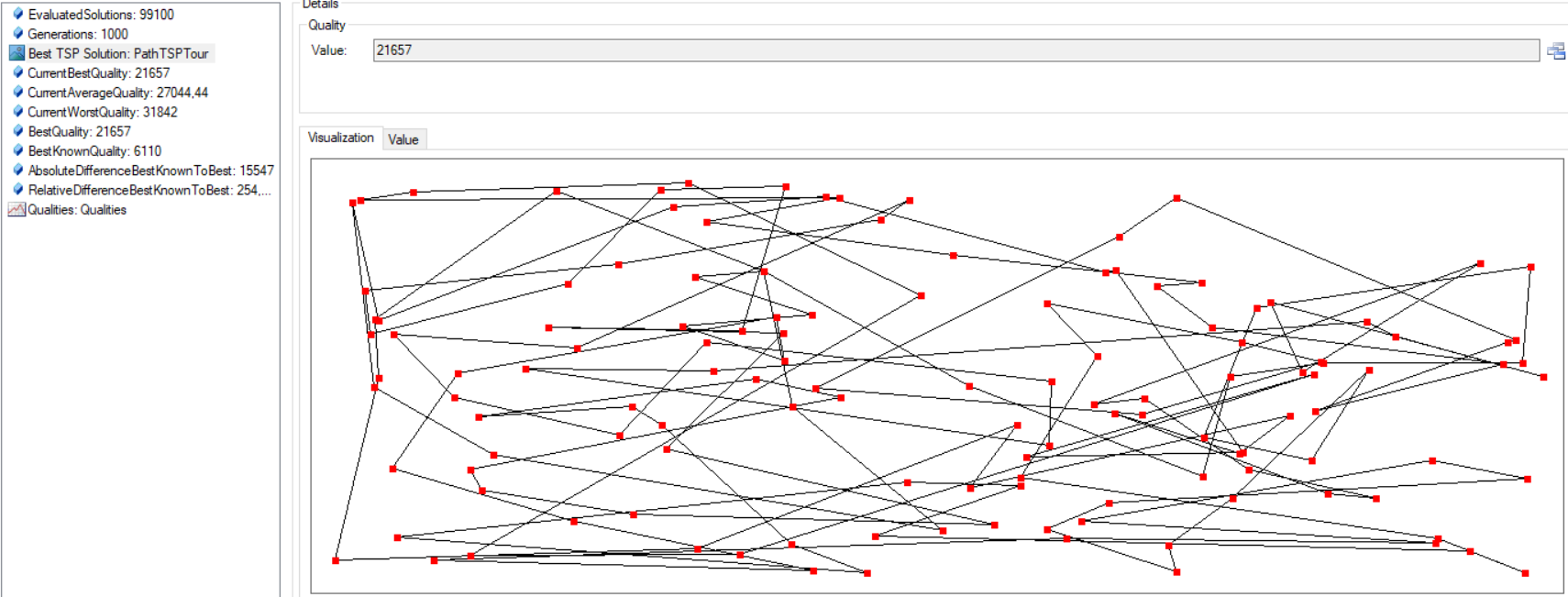
MutationProbability: 50%

PopulationSize: 100

Crossover: OrderCrossover2

Mutator: InversionManipulator

Selector: ProportionalSelector



**3.7 Rulare cu probabilitate de mutație 95%**

Elites: 1

MaximumGenerations: 1000

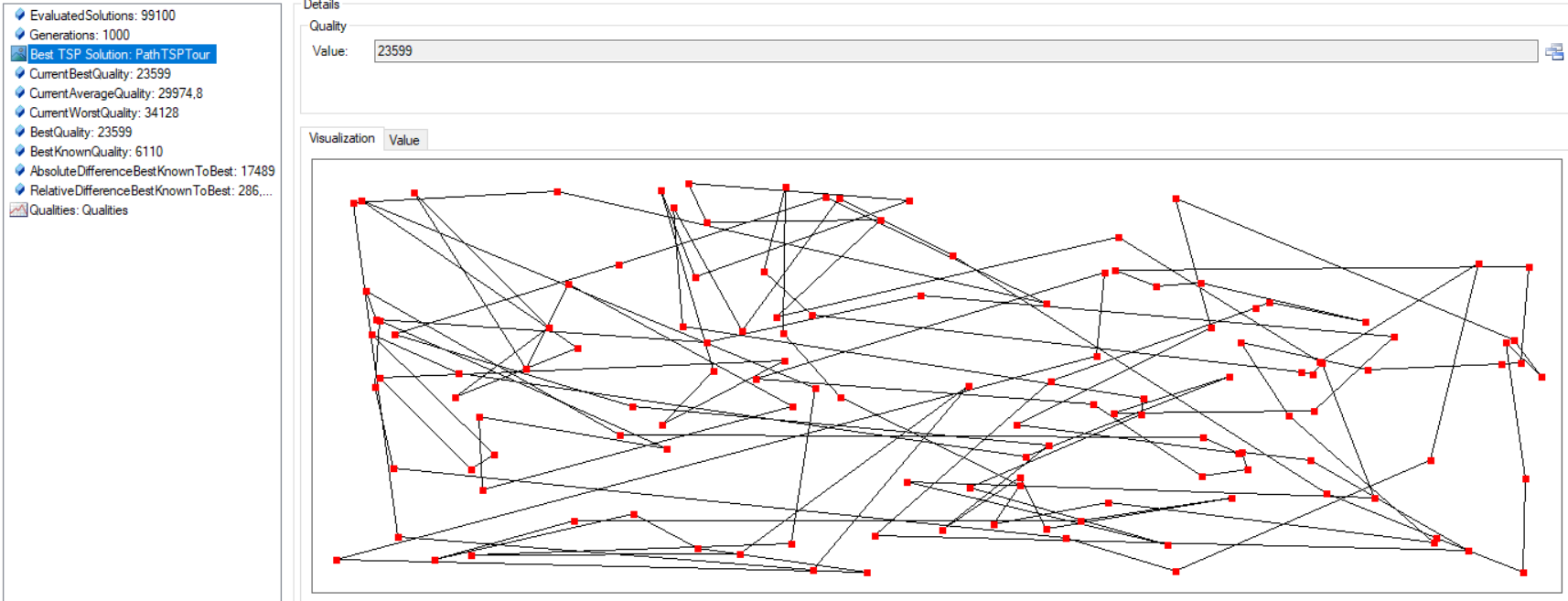
MutationProbability: 95%

PopulationSize: 100

Crossover: OrderCrossover2

Mutator: InversionManipulator

Selector: ProportionalSelector



**3.8 Rulare cu populație maximă 50**

Elites: 1

MaximumGenerations: 1000

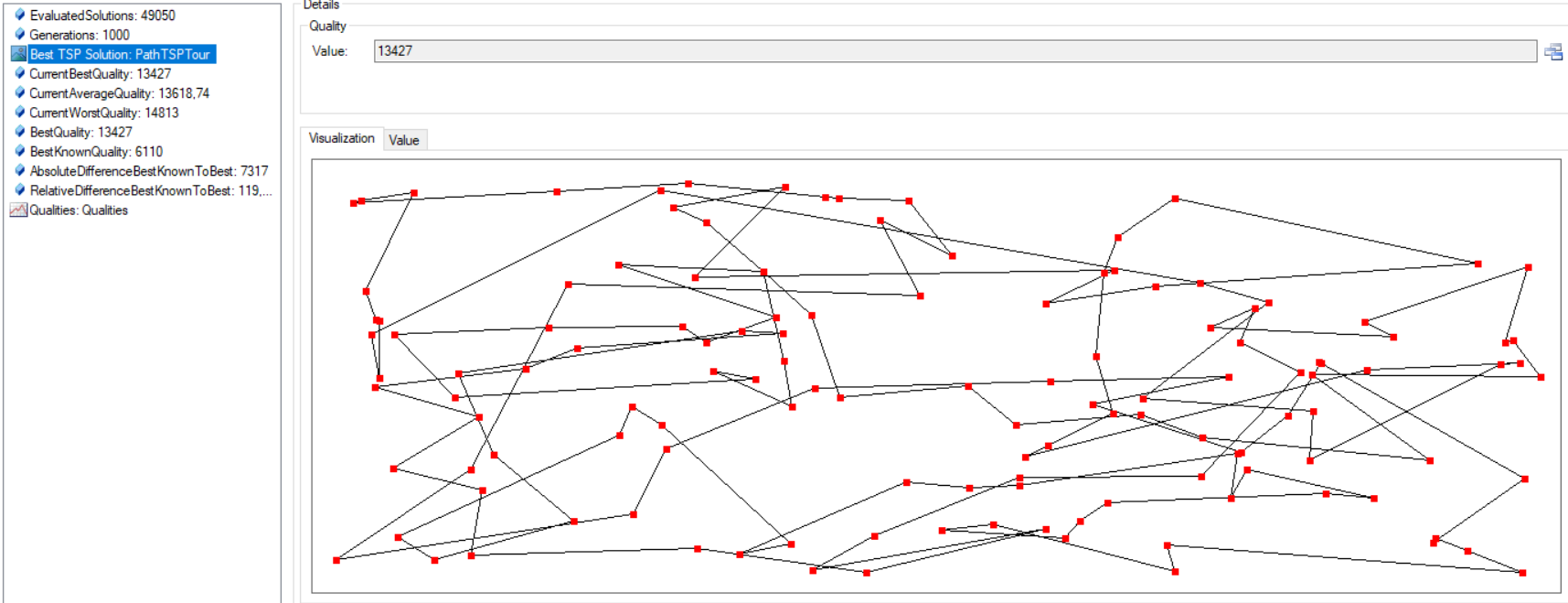
MutationProbability: 5%

PopulationSize: 50

Crossover: OrderCrossover2

Mutator: InversionManipulator

Selector: ProportionalSelector



**3.9 Rulare cu populație maximă 500**

Elites: 1

MaximumGenerations: 1000

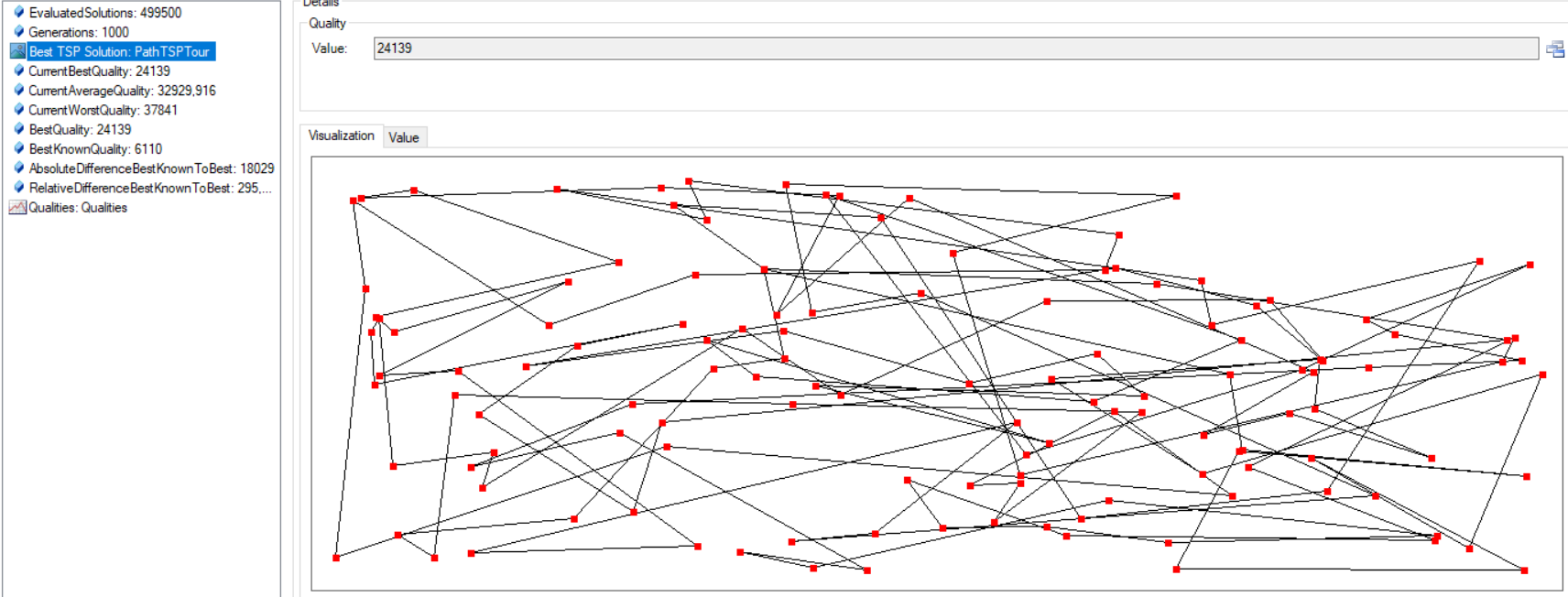
MutationProbability: 5%

PopulationSize: 500

Crossover: OrderCrossover2

Mutator: InversionManipulator

Selector: ProportionalSelector



**3.10 Rulare cu populație maximă 1000**

Elites: 1

MaximumGenerations: 1000

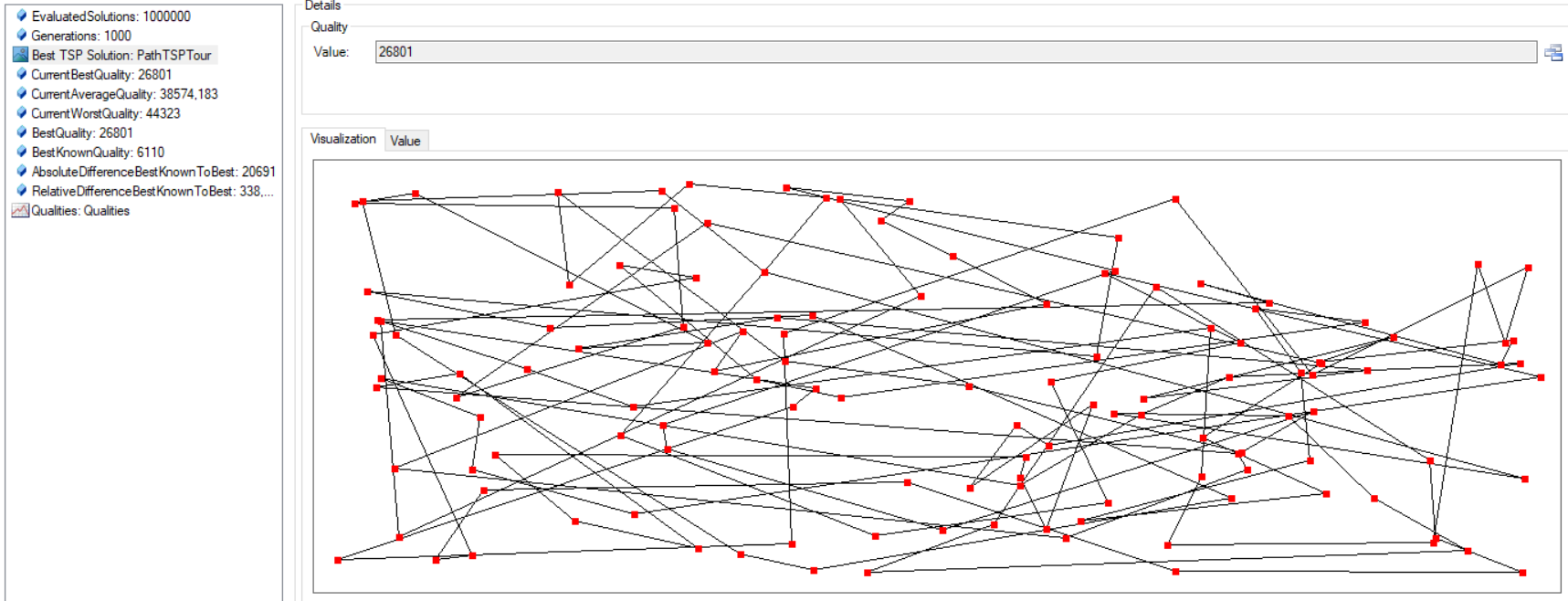
MutationProbability: 5%

PopulationSize: 1000

Crossover: OrderCrossover2

Mutator: InversionManipulator

Selector: ProportionalSelector



**3.11 Rulare cu 0 membrii de elită**

Elites: 0

MaximumGenerations: 1000

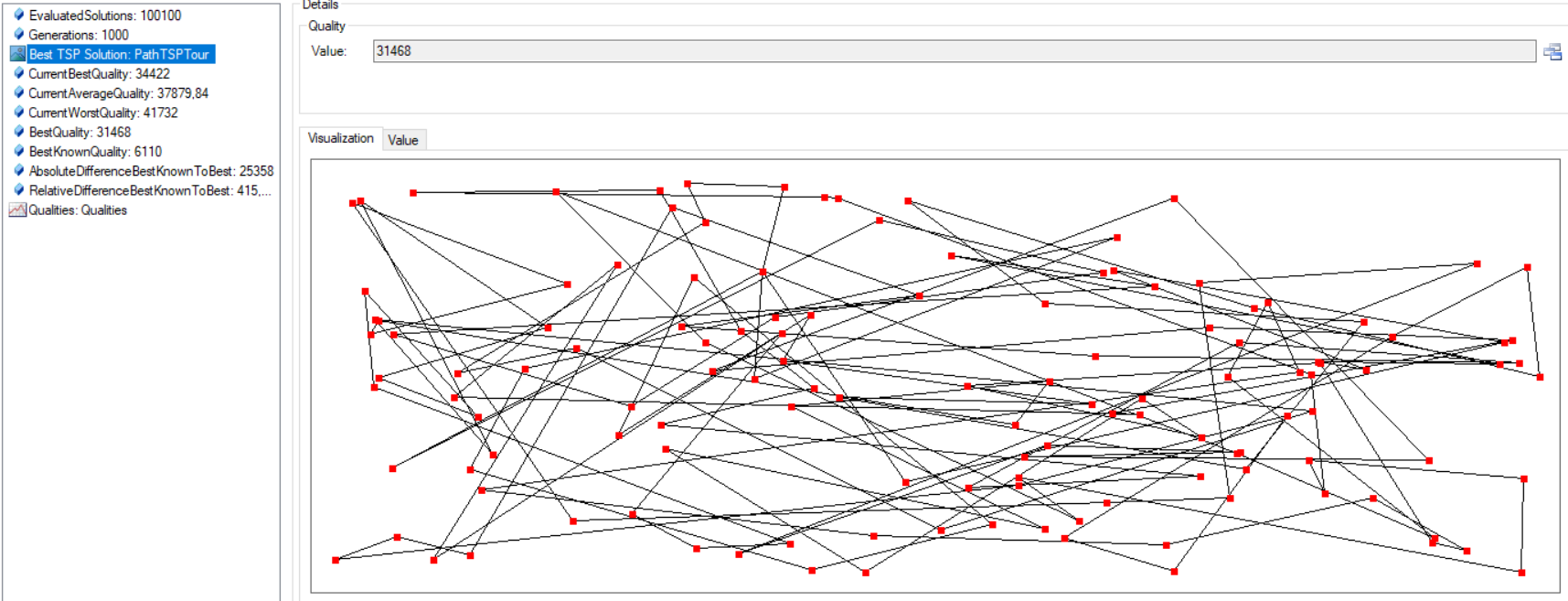
MutationProbability: 5%

PopulationSize: 100

Crossover: OrderCrossover2

Mutator: InversionManipulator

Selector: ProportionalSelector



**3.12 Rulare cu 5 membrii de elită**

Elites: 5

MaximumGenerations: 1000

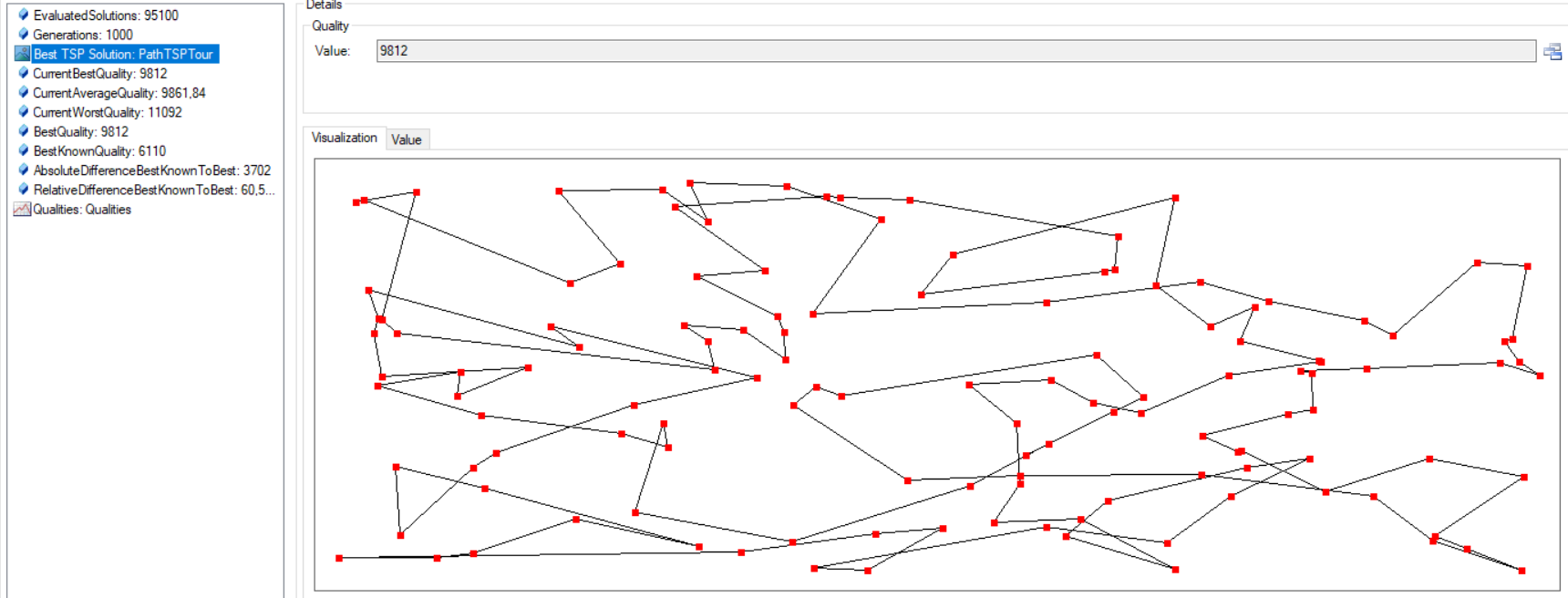
MutationProbability: 5%

PopulationSize: 100

Crossover: OrderCrossover2

Mutator: InversionManipulator

Selector: ProportionalSelector



**3.13 Rulare cu 10 membrii de elită**

Elites: 10

MaximumGenerations: 1000

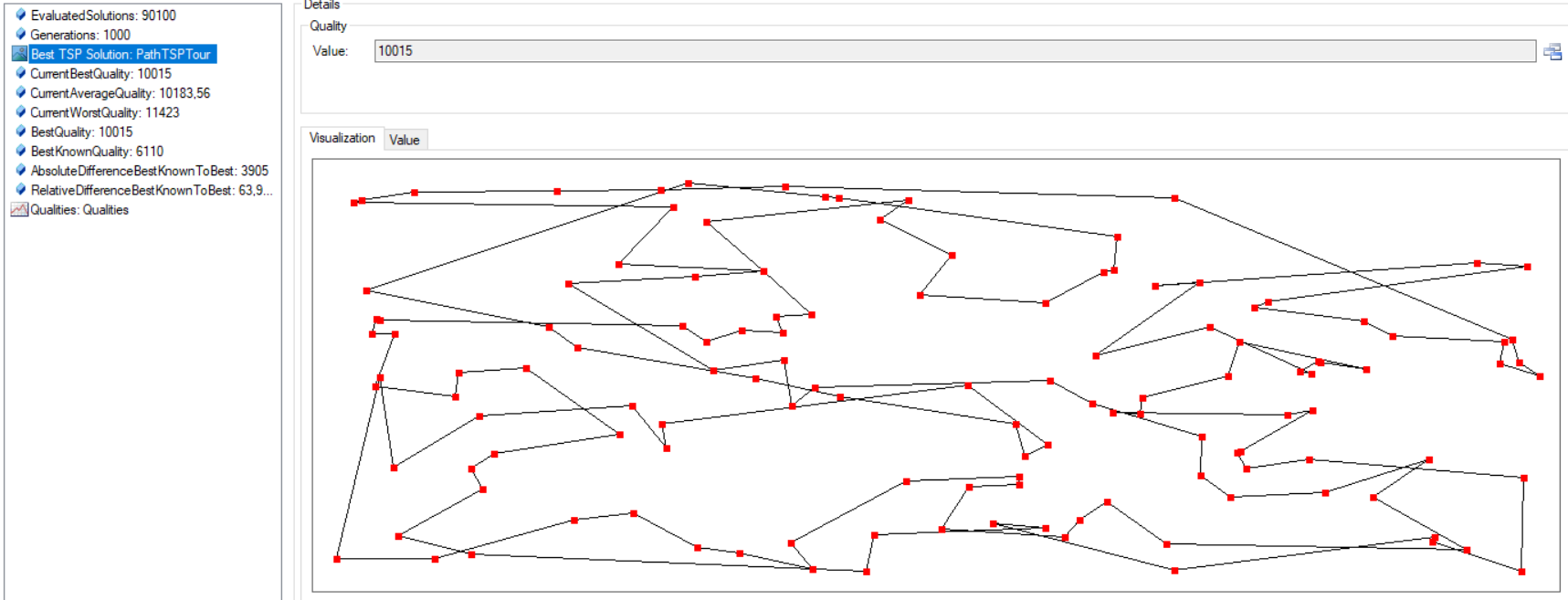
MutationProbability: 5%

PopulationSize: 100

Crossover: OrderCrossover2

Mutator: InversionManipulator

Selector: ProportionalSelector



**3.14 Rulare cu selector BestSelector**

Elites: 1

MaximumGenerations: 1000

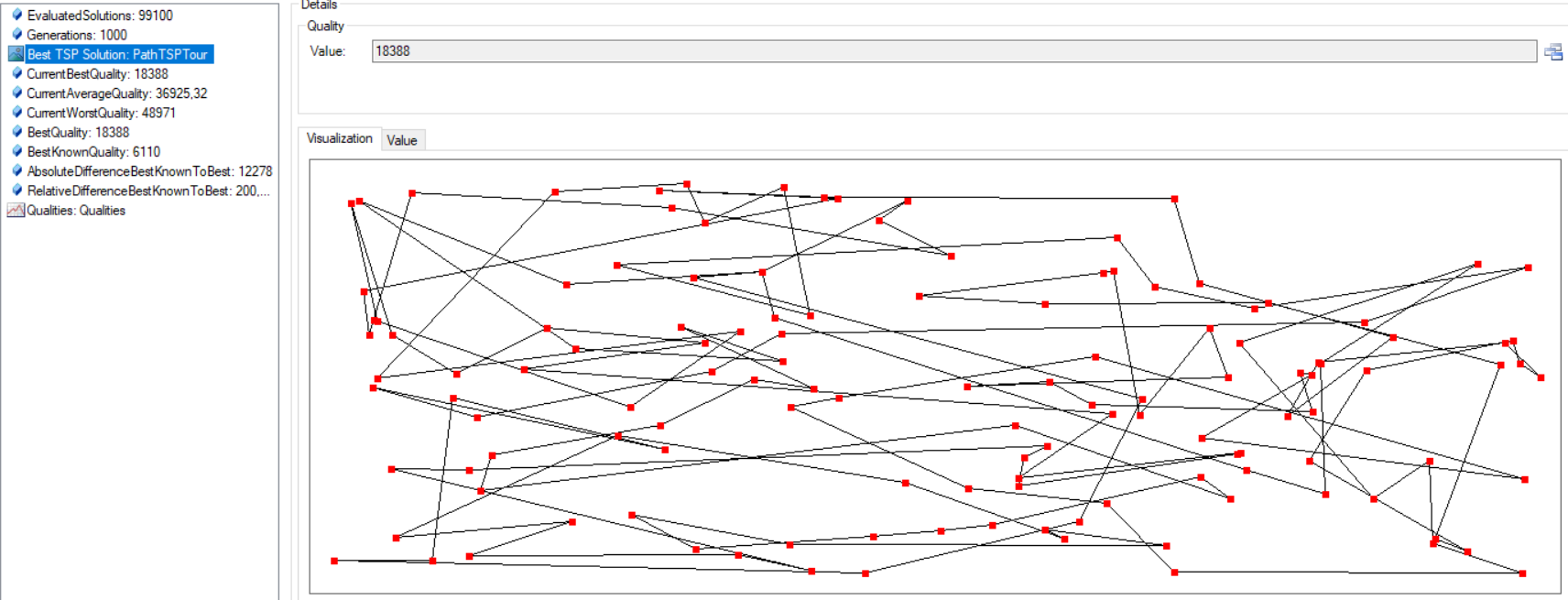
MutationProbability: 5%

PopulationSize: 100

Crossover: OrderCrossover2

Mutator: InversionManipulator

Selector: BestSelector



**3.15 Rulare cu selector GenderSpecificSelection**

Elites: 1

MaximumGenerations: 1000

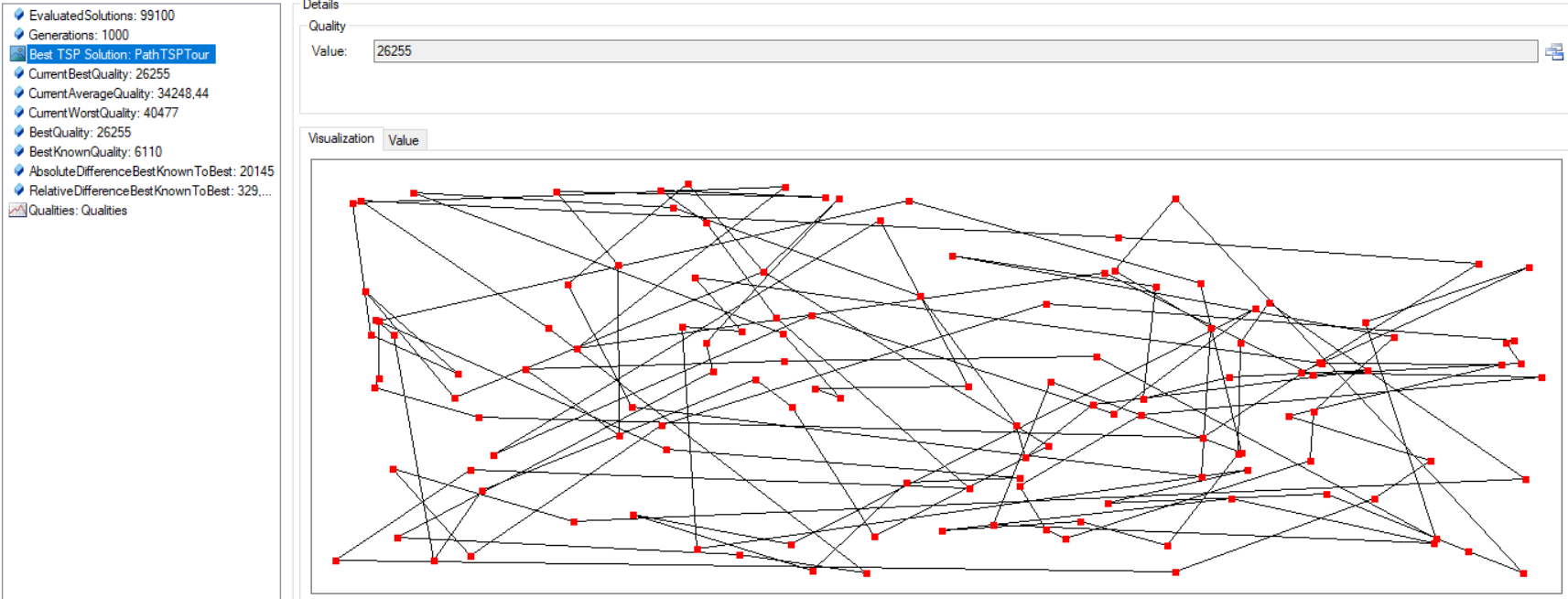
MutationProbability: 5%

PopulationSize: 100

Crossover: OrderCrossover2

Mutator: InversionManipulator

Selector: GenderSpecificSelection



**3.16 Rulare cu selector RandomSelector**

Elites: 1

MaximumGenerations: 1000

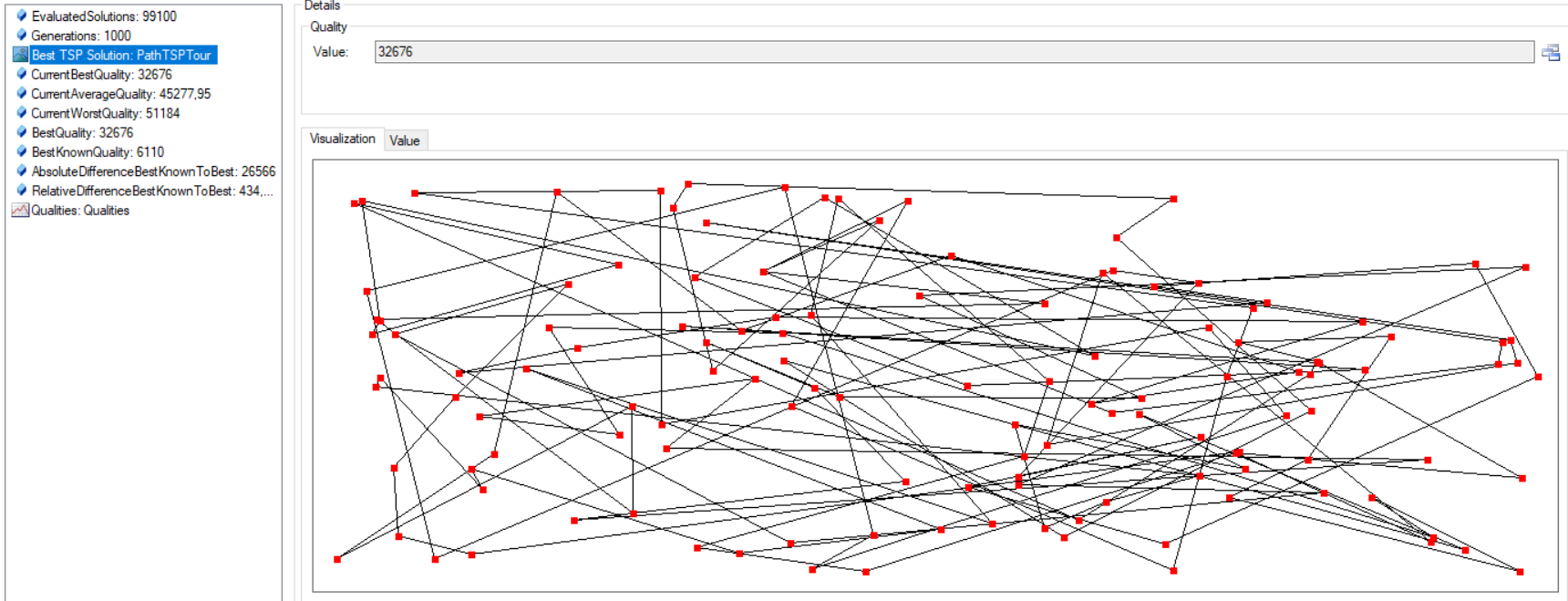
MutationProbability: 5%

PopulationSize: 100

Crossover: OrderCrossover2

Mutator: InversionManipulator

Selector: RandomSelector



**3.17 Rulare cu generație maximă 10000, 50 membrii de elită și selector** **GeneralizedRankSelector**

Elites: 50

MaximumGenerations: 10000

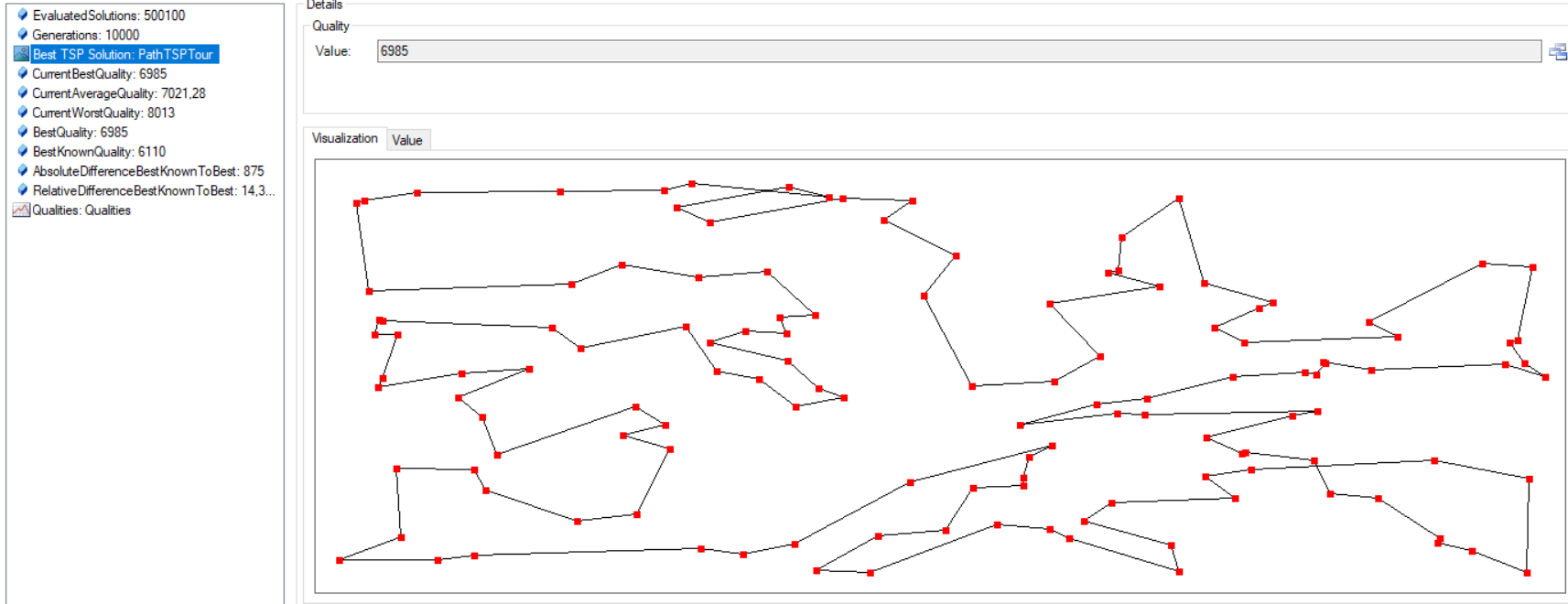
MutationProbability: 5%

PopulationSize: 100

Crossover: OrderCrossover2

Mutator: InversionManipulator

Selector: GeneralizedRankSelector



4. Concluzii

* Cu cât crește generația maximă, cu atât crește eficiența găsirii drumului optim, dar și durata de execuție a algoritmului;
* Cele mai optime drumuri se obțin cu o probabilitate de mutație în jur de 5%, iar cum crește cu atât se ajunge la un drum mai neoptim;
* Cu cât crește numărul membrilor de elită, cu atât crește eficiența găsirii drumului optim
* Selectorul random produce rezultate diferite de fiecare dată când este folosit;
* Rezultate optime se obțin cu o populație de 100 de indivizi per generație;