

Serverio resursų optimizavimas

Vygintas Kazanavičius, IFM-9/1, e-paštas: vkaza@ifko.ktu.lt

Uždavinys

Serveryje yra N vietų plokštėms įstatyti, tačiau to paties tipo plokštės nebūtinai galima statyti į visas N vietas. Todėl kiekvienai vietai i apibrėžiama galimų plokščių resursų aibė:

$$R_i = \{r_{i,1}, r_{i,2}, \dots, r_{i,N_i}\}$$

Tuomet į vietą i įstatytos plokštės resursas bus $r_i \in R_i$.

Šiame darbe maksimizuojamas serverio suminis resursas R , kuris apibrėžiamas kaip serveryje įstatytų plokščių resursų r_i suma:

$$R = \sum_{i=1}^N r_i,$$

beto reikalaujama, kad serverio suminis resursas R neviršytų serverio pajėgumo M (maksimaliai leistino resurso):

$$(R = \sum_{i=1}^N r_i) \leq M.$$

Tokio tipo uždavinys yra ypač aktualus telekomunikacijų paslaugų tiekėjams ir gali būti taikomas įvairiems panaudojimo atvejams:

1. Kaip optimaliai sukomplektuoti mazgo tinklinę įrangą, kai nupirkta tuščias blokas?
2. Ar esama bloko konfigūracija optimali?
3. Kaip optimaliai vykdyti tinklo plėtrą? Siūlomo metodo pagalba būtų galima sudaryti instrukcijas nurodančias kur reikia statyti tam tikrą naują plokštę, kad plečiant tinklą būtų efektyviai išnaudojami serverio pajėgumai.

Šiame darbe uždavinys suvedamas prie globalaus minimizavimo uždavinio apibrėžiant tikslo funkciją $F(x)$ kaip serverio pajėgumų M išnaudojimo efektyvumą:

$$F(\vec{x}) = M - \sum_{i=1}^N r_{i,x_i}, \text{ čia}$$

$\vec{x} = (x_1, x_2, \dots, x_i, \dots, x_N)$ - serverio konfigūracijos vektorius, kur x_i yra i -joje vietoje įstatytos plokštės tipo numeris, o tos plokštės resursas - $r_{i,x_i} \in R_i$.

Optimizavimo metodai

Šiame darbe resursų optimizavimui buvo realizuoti šie optimizavimo metodai:

1. Monte Carlo
2. Simulated Annealing (SA)
3. SA parametrų optimizavimas GMJ paketo metodais (Bayes, MC, SA, Globt, Exkor).

Duomenų failo formatas

```
<Vietų kiekis serveryje - N>  
<Maksimalus serverio plokščių suminis resursas, kurio negalima viršyti - M>  
<1 vietos 1 plokštės resursas g[1][1]><1 vietos 1 plokštės resursas g[1][2]>...  
<2 vietos 1 plokštės resursas g[2][1]><2 vietos 1 plokštės resursas g[2][2]>...  
...  
<n vietos 1 plokštės resursas g[n][1]><n vietos 1 plokštės resursas g[n][2]>...
```

Pvz.: jeigu serveryje yra $N=6$ vietos plokštėms statyti, o serverio maksimaliai leistinas suminis resursas $M=95$, o serverio vietose galima statyti plokštes kurių resursai atitinkamai yra:

Vietos nr.	Galimi resursai
1	16 14 12 9 5 3 1 0
2	26 12 5 3 1 0
3	16 14 12 9 5 3 1 0
4	14 9 5 3 1 0
5	16 14 12 9 5 3 1 0
6	16 12 9 5 3 1 0

Pastabos:

- Galimi resursai užrašomi eilute skiriant tarpu ir rašomi mažėjimo tvarka.
- Jeigu tam tikroje vietoje galima plokštės nestatyti, tai resursų eilutės gale rašomas 0

Tuomet duomenų failas atrodys taip:

```
6  
95  
16 14 12 9 5 3 1 0  
26 12 5 3 1 0  
16 14 12 9 5 3 1 0  
14 9 5 3 1 0  
16 14 12 9 5 3 1 0  
16 12 9 5 3 1 0
```

Darbo su programa scenarijai

Optimizavimas Monte Carlo metodu

- Method lange pasirinkite MonteCarlo
- Nurodykite iteracijų kiekį:

Method Task Operation

Select method and properties: MonteCarlo

Property	Value
Iterations	500

3. Pasirinkite Task->Cards
4. Pasirinkite pradinio taško parinkimo strategiją (Greedy – gobši, Random – atisiktinė)
5. Nurodykite duomenų failo pilną internetinį adresą:

Method Task Operation

Select task and properties: Cards

Property	Value
Initial point	Greedy point
Data file URL	http://ifko.ktu.lt/~vkaza/cards/lib/plokstes.dat

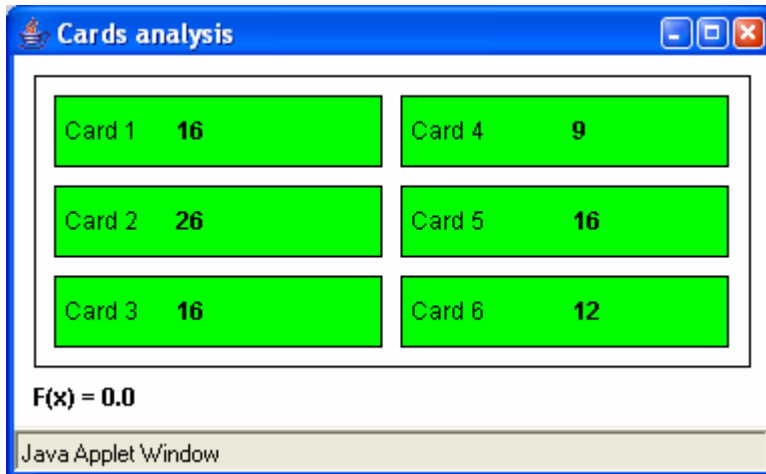
6. Spauskite Operation->Run
7. Ekrane matysite rezultatus:

Method Task Operation

Property	Value
Iteration	690
F(x)	0.0
Card 1	16.0
Card 2	26.0
Card 3	16.0
Card 4	9.0
Card 5	16.0
Card 6	12.0

Run Pause Stop Analysis

8. Jei norite pamatyti optimalų plokščių išdėstymą serveryje, spauskite Analysis->CardsAnalyzer:

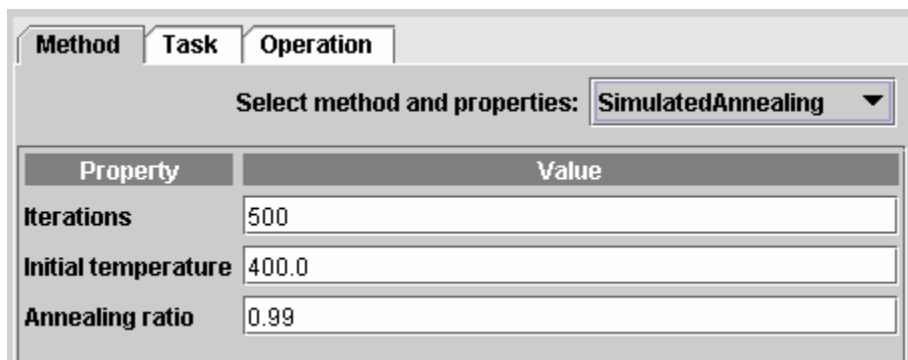


Optimizavimas Simulated Annealing metodu

1. Method lange pasirinkite SimulatedAnnealing
2. Nurodykite iteracijų kiekį
3. Nurodykite pradinę temperatūrą lauke initial temperature
4. Nurodykite šildymo greitį lauke annealing ratio

Pastaba sekančios iteracijos temperatūra šiame metode apskaičiuojama pagal formulę:

$$T(k+1) = T(k) * annealing_ratio$$



Toliau vykdykite MonteCarlo 3-8 etapus.

SA parametru optimizavimas

1. Method lange pasirinkite bet kurį metodą prasidedantį raide W (WBayes, WMonteCarlo, ir t.t.)
2. Nustatykite metodo parametrus

Method Task Operation

Select method and properties: **WBayes**

Property	Value
Iterations	100
Initial points	5

3. Pasirinkite Task->SAParamOptimizer
4. Nustatykite parametrus kaip Simulated Annealing Cards Task atveju
5. Lauke No of iterations during 1 SA nurodykite kiek reikia atlikti iteracijų darant vieną Simulated Annealing etapą
6. Nurodykite SA parametrų kitimo ribas:

Method Task Operation

Select task and properties: **SAParamOptimizer**

Property	Value		
Initial point	Greedy point		
Data file URL	http://ifko.ktu.lt/~vkaza/cards/lib/plokstes.dat		
No of iterations during 1 SA	500		
SA Averaging factor N	50		
Dimension	Min	Default	Max
Initial temperature (x1)	200.0	200.0	40000.0
Annealing ratio (x2)	0.1	0.1	1.0

7. Operation->Run
8. Ekrane matysite surastus optimalius SA parametrus ir $F(x)$ minimalią reikšmę prie šių parametrų:

Method Task Operation

Property	Value
Iteration	99
F(x)	0.0
Initial temperature	5175.0
Annealing ratio	0.438

9. Kaip ir visuose metoduose galite pažiūrėti optimalų plokščių išdėstymą prie optimalių parametrų per Analysis->CardsAnalyzer